# STATISTICAL LEARNING FOR STRUCTURAL PATTERNS WITH TREES

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Xiaohan Yan

August 2018

STATISTICAL LEARNING FOR STRUCTURAL PATTERNS WITH TREES

Xiaohan Yan, Ph.D.

Cornell University 2018

In achieving structural patterns in parameters, we focus on two challenging cases in which (1) hierarchical sparsity pattern is desired such that one group of parameters is set to zero whenever another is set to zero; and (2) many features that are counts of rarely occurring events are present, and appropriate aggregation of the rare features may lead to better estimation. In either case, the methods under consideration use a tree or a directed acyclic graph (DAG) that encodes relations among parameters as side information.

For achieving hierarchical sparsity patterns in parameters, we investigate the differences between group lasso (GL) and latent overlapping group lasso (LOG) in terms of their statistical properties and computational efficiency. We highlight a phenomenon of GL in which parameters embedded deep within the DAG are more aggressively regularized than those that are less deeply embedded. By contrast, we show that using LOG fulfills our goal without any additional complication and performs, both in practice and in theory, very similarly to the GL penalty that is modified to curb its over-aggressiveness. In terms of computation, we derive a finite-step algorithm for the proximal operator of LOG in the case of the DAG being a directed path graph; we later exploit this efficiency to propose a novel path-based block coordinate descent scheme. Finally, we compare the two frameworks in estimating banded covariance matrix, where we introduce a new sparsely-banded estimator using LOG, which we show achieves the statistical advantages of an existing GL-based method but is

simpler to express and more efficient to compute.

Another kind of sparsity we care about is sparsity in the data itself. It is prevalent to have many highly sparse features for counting frequency of rare events in diverse areas, ranging from natural language processing (e.g., rare words) to biology (e.g., rare species). We show, both theoretically and empirically, that not explicitly accounting for the rareness of features can greatly reduce the effectiveness of an analysis. We propose a tree-guided framework for aggregating rare features into denser ones through solving a convex optimization problem. The tree, which encodes feature similarity information on the leaves, comes from prior knowledge or data sources external to the current problem and is used as side information in aggregation. In our proposal, aggregating rare features is equivalent as enforcing equal coefficients within each group learned from solving the convex problem, resulting in another case of structural pattern in parameters. We apply our method on two data sets: a TripAdvisor hotel review data set, in which we predict the numerical rating of a hotel based on the text of an associated review; and a microbiome data set from the American Gut project that measures microbial species abundance from fecal samples, in which we predict the one's BMI based on both microbiome and non-microbiome features. In both applications, our method achieves high accuracy by making effective use of rare features and yields more interpretable results.

## BIOGRAPHICAL SKETCH

Xiaohan Yan was born and grew up in Taian, Shandong Province, China. He received his undergraduate degrees from University of Washington in Statistics, Economics and Applied Mathematics. While at University of Washington, he won a Mary Gates Research Scholarship for his undergraduate research study on China's energy efficiency. After graduating in 2013, Xiaohan entered Cornell University to pursue a Ph.D. in Statistics. During his graduate study, Xiaohan worked on developing statistical learning methods that are scalable for high dimensions, and became passionate in data science through research projects and industry experiences. Following the completion of his Ph.D., Xiaohan will join Microsoft as a Data Scientist.

To my parents

## ACKNOWLEDGEMENTS

First and foremost, I would like to thank my Ph.D. advisor, Jacob Bien for his guidance, time and commitment to my development as an independent researcher. His deep statistical knowledge, creativity and elegant presentations have been inspirational. Jacob has been an amazing academic advisor, a colleague and often a life coach. During my last year of graduate study, Jacob spent a great deal of time and effort helping me to get a job, going as far as having skype meeting with me on Saturday morning to work on my job talk and coaching me on interview skills. He has been very supportive of my research interests and career decisions, and his patience with me is beyond my expectation. Needless to say, I feel very fortunate to have had Jacob as my Ph.D. advisor.

Next, I thank my committee members, Martin T. Wells and Thorsten Joachims for always providing sound advice and deep insight. Martin motivated me with his enthusiasm towards research and provided me crucial support several times in my graduate study. Thorsten opened the door for me to see machine learning from a computer scientist's perspective. I would like to express my sincere gratitude to Martin and Thorsten.

Christian L. Müller from the Simons Foundation tolerated my poor background of biology, and contributed his expertise in computational biology in our collaboration. Without him Chapter 4 of the thesis would not be possible. I am grateful for his sound advice and generosity with his time.

I owe a debt of gratitude to my undergraduate advisors, Galen R. Shorack and June G. Morita, for invigorating my interest in statistics and encouraging me to pursue graduate studies.

My fellow Ph.D. students and friends, especially William B. Nicholson, David Sinclair, Guo (Hugo) Yu, Ze Jin, Zi Ye, Xiaoyun Quan and Ben Baer, have

been a great source of support, encouragement, commiseration and joy. They made my Ph.D. a wonderful memory and I appreciate their companionship.

My gratitude goes out to my parents for emphasizing the value of higher education and supporting me through this long journey. My father provided me advice at several important points in my life, and he was proven right most of the time. My mother taught me to hold an active and optimistic attitude towards life. I love them very much and could not be in the U.S. pursuing my dreams without them.

Finally, I especially thank my wife Tianlu Xu for her unwavering support during my Ph.D. and taking great care of our daughter Keyu. I thank her from the bottom of my heart.

## TABLE OF CONTENTS


Biographical Sketch . . . iii
Dedication . . . iv
Acknowledgements . . . v
Table of Contents . . . vii
List of Tables . . . ix
List of Figures . . . x

**1 Introduction** **1**

**2 A Choice of Two Regularizers in Hierarchical Sparse Modeling** **4**
2.1 Introduction . . . 4
2.2 Hierarchical Sparse Modeling: Two Frameworks . . . 9
2.2.1 The Group Lasso Approach . . . 10
2.2.2 The Latent Overlapping Group Lasso Approach . . . 12
2.2.3 Are These Two Approaches Different? . . . 13
2.3 Differential Shrinkage of GL . . . 15
2.4 Computation . . . 20
2.4.1 Naive BCD for LOG . . . 22
2.4.2 Solution of the LOG Prox for a Directed Path Graph . . . 24
2.4.3 Path-Based BCD and ADMM for LOG . . . 26
2.5 Estimating Banded Covariance with LOG . . . 30
2.5.1 Defining the Estimator $\hat{\mathbf{\Sigma}}^{\text{LOG}}$ . . . 32
2.5.2 Statistical Properties of $\hat{\mathbf{\Sigma}}^{\text{LOG}}$ . . . 33
2.5.3 Simulation Study . . . 35
2.6 Conclusion . . . 39

**3 A Tree-Based Rare Feature Selection Framework in High Dimensions** **41**
3.1 Introduction . . . 41
3.2 Rare Features and the Promise of Aggregation . . . 46
3.2.1 The Difficulty Posed by Rare Features . . . 46
3.2.2 Aggregating Rare Features Can Help . . . 48
3.3 Main Proposal: Tree-Guided Aggregation . . . 50
3.3.1 A Tree to Guide Aggregation . . . 50
3.3.2 A Tree-Based Parametrization . . . 51
3.3.3 The Optimization Problem . . . 53
3.4 Statistical Theory . . . 56
3.5 Simulation Study . . . 58
3.6 Application to Hotel Reviews . . . 61
3.7 Conclusion . . . 67

**4 Microbiome Compositional Feature Selection with Phylogenetic Tree 68**
4.1 Introduction . . . 68
4.2 Model and Method . . . 73
4.2.1 The Log-Contrast Model . . . 73
4.2.2 Phylogenetic-Tree-Guided Aggregation . . . 74
4.3 Simulation Study . . . 79
4.4 Gut Microbiome Analysis . . . 84
4.5 Conclusion . . . 91

**5 Conclusions 93**

**A Appendix for Chapter 2 95**
A.1 Proof of Lemma 1 . . . 95
A.2 Proof of Propositions 1 and 2 . . . 96
A.2.1 Proof of Proposition 1 . . . 96
A.2.2 Proof of Proposition 2 . . . 98
A.3 Proof that Algorithm 3 Solves $\text{Prox}_{\text{LOG}}^{a(\mathcal{D})}$ for a Directed Path Graph . 101
A.4 Computational Complexity of Algorithm 3 . . . 107
A.5 Computational Complexity of GL for a Directed Path Graph . . . 108
A.5.1 GL Proximal Operator . . . 108
A.5.2 Modified GL Proximal Operator . . . 109
A.6 Proof of Lemma 4 . . . 111
A.7 Simple Algorithm for Path Decomposition of DAG . . . 112
A.8 Proof of Lemma 5 . . . 112
A.9 Proof of Theorem 1 . . . 114
A.10 Proof of Theorem 2 . . . 116
A.11 Proof of Theorem 3 . . . 117
A.12 Algorithm 8 for Solving Problem (2.20) . . . 118
A.13 PSD Probability (Figure A.1) and Minimum Eigenvalues (Figure A.2) of the Three Covariance Estimators . . . 118

**B Appendix for Chapter 3 121**
B.1 Failure of OLS in the Presence of A Rare Feature . . . 121
B.2 Proof of Theorem 5 . . . 121
B.3 Consensus ADMM for Solving Problem (3.5) . . . 126
B.3.1 Derivation of Algorithm 5 . . . 126
B.3.2 Treatment of Intercept in Problem (3.5) . . . 131
B.4 Proof of Lemma 6 . . . 131
B.5 Proof of Theorem 6 . . . 132
B.6 Proof of Corollary 1 . . . 134

**C Appendix for Chapter 4 136**
C.0.1 LA-ADMM for Solving the Consensus Probelm in (4.5) . . 136
C.0.2 Treatment of Environemental Covariates in (4.4) and (4.2) 141

## LIST OF TABLES

2.1 Applications of GL and LOG in HSM . . . . . 7

3.1 Performance of five methods on the held-out test set: `L1` is the lasso; `L1-dense` is the lasso on only dense features; `L1-ag-h` is the lasso with features aggregated based on height; and `L1-ag-d` is the lasso with features aggregated based on density level. . . . . 64

3.2 Term density and estimated coefficient for adjectives in the selected group . . . . . 66

4.1 Test set MSE and model size for CV-chosen fit and best-performing fit . . . . . 86

4.2 Aggregations recovered by our method at its best performance: *Aggreg. Level* is the taxonomic level at which an aggregation occurs, and *Aggregated Taxon* is the corresponding taxon at aggregation; *Size of Aggreg.* is the number of OTUs in an aggregation; $\hat{\boldsymbol{\beta}}^{\text{ours}}$ is the estimated shared coefficient for all OTUs in an aggregation; *OTU Density L.B.* is the density of the rariest OTU in an aggregation; *OTU Density U.B.* is the density of the most abundant OTU in an aggregation; *Aggreg. Density* is the density of an aggregation; *No. Shared OTUs (lasso)* is number of OTUs in an aggregation that are also selected by (4.2) at OTU level. Density is percentage of samples containing an OTU. . . . . 90

4.3 For the 11 OTUs associated with Akkermansia muciniphila, their densities (in percentage) and estimated regression coefficients from our method ($\hat{\boldsymbol{\beta}}^{\text{ours}}$), (4.2) at OTU level ($\hat{\boldsymbol{\beta}}^{\ell_1,\text{OTU}}$) and (4.2) at genus level ($\hat{\boldsymbol{\beta}}^{\ell_1,\text{genus}}$). . . . . 91

## LIST OF FIGURES


2.1 (Left) A DAG $\mathcal{D}$ for a two-way interaction model with three predictors. In HSM, the DAG $\mathcal{D}$ encodes the sparsity structure: a node's parameters must be set to zero if it has a parent with zeroed parameters. (Right) The same $\mathcal{D}$ specified using our notation: each node contains only one element and the correspondence between $s_i$ and $\beta_j$ is as shown. In red dashed contour, ancestors$(\mathcal{D}; s_5) = \{s_1, s_3, s_5\}$ include both main effects, $\beta_1$ and $\beta_3$, in the ancestor group of the interaction effect $\beta_{13}$. In blue solid contour, descendants$(\mathcal{D}; s_2) = \{s_2, s_4, s_6\}$ contains both interaction effects involving main effect $\beta_2$. . . . . 10

2.2 For the same DAG as in Figure 2.1, an illustration of group structures $\mathcal{G} = d(\mathcal{D})$ and $\mathcal{G} = a(\mathcal{D})$ induced for GL and LOG, respectively. (Top) The group structure $d(\mathcal{D})$ for GL is shown in solid contours: $d(\mathcal{D}) = \{s_4, s_5, s_6, s_2 \cup s_4 \cup s_6, s_1 \cup s_4 \cup s_5, s_3 \cup s_5 \cup s_6\}$. Each group of $d(\mathcal{D})$ can be thought of as a set of the efffect itself and all the relevant interaction effects. (Bottom) The group structure $a(\mathcal{D})$ for LOG is shown in dashed contours: $a(\mathcal{D}) = \{s_1, s_2, s_3, s_1 \cup s_3 \cup s_5, s_1 \cup s_2 \cup s_4, s_2 \cup s_3 \cup s_6\}$. Each group of $a(\mathcal{D})$ can be described as a set of the effect itself and all the relevant main effects. . . . . 11

2.3 Directed Path Graph with $D$ Nodes . . . . . 11

2.4 For $\beta \in \mathbb{R}^3$ and the DAG $\{1\} \to \{2\} \to \{3\}$, (Left) the unit ball of $\Omega_{\text{GL}}^{d(\mathcal{D})}(\beta; w)$ where $d(\mathcal{D}) = \{\{1, 2, 3\}, \{2, 3\}, \{3\}\}$ and $w = (1, 1, 1)$ and (Right) the unit ball of $\Omega_{\text{LOG}}^{a(\mathcal{D})}(\beta; w)$ where $a(\mathcal{D}) = \{\{1\}, \{1, 2\}, \{1, 2, 3\}\}$ and $w = (1, \sqrt{2}, \sqrt{3})$. . . . . 13

2.5 The effect of the proximal operator of three regularizers on $\beta_i^* = 1 - \frac{i-1}{D}$: (Left) $\hat{\beta}^{\text{GL}}$, (Middle) $\hat{\beta}^{\text{LOG}}$ and (Right) $\hat{\beta}^{\text{mGL}}$. . . . . 18

2.6 The effect of the proximal operator of three regularizers on $\beta_i^* = 1_{\{i \le D/2\}} + 0.5 * 1_{\{i > D/2\}}$: (Left) $\hat{\beta}^{\text{GL}}$, (Middle) $\hat{\beta}^{\text{LOG}}$ and (Right) $\hat{\beta}^{\text{mGL}}$. . 20

2.7 Let $s_i = \{i\}$ for $i \in \{1, \cdots, 8\}$. (Left) $a(\mathcal{D})$ is decomposed into 3 path graphs: $\mathcal{P}^{(1)}$ (in green solid contour), $\mathcal{P}^{(2)}$ (in red dashed contour) and $\mathcal{P}^{(3)}$ (in blue dotted contour). (Middle) The partition of $\mathcal{G} = a(\mathcal{D})$: $\mathcal{G}_1$, $\mathcal{G}_2$ and $\mathcal{G}_3$ (colored accordingly). (Right) $a(\mathcal{D})$ can be thought of as three separate path graphs on a new set of nodes, with parameter assignments shown inside each node: (in green solid contour) $supp(\beta^{(1)}) \subseteq \{1, \cdots, 7\}$, (in red dashed contour) $supp(\beta^{(2)}) \subseteq \{3, 4, 5, 6, 8\}$ and (in blue dotted contour) $supp(\beta^{(3)}) \subseteq \{3, 5\}$. . . . . 27

2.8 (Top) Tree structures for example 1, 2 and 3, respectively. On top left, $T_1$ and $T_2$ are path graphs of length 50 and 49, respectively. (Bottom) Plot of ratio of the difference in objective values of the two BCDs and the difference in objective value of the path-based BCD and the "truth", evaluated at each cycle and averaged over 20 realizations, with the corresponding tree above it. . . . . 28
2.9 (Left) The group $s_{1:2}$; (Right) The nested groups of the form $s_{1:k}$ in $a(\mathcal{D})$. . . . . 33
2.10 (Left) $MSE(\lambda_{theory})$ and (Right) $MSE(\lambda_{theory})/\log p$ as a function of $K$ for $\hat{\boldsymbol{\Sigma}}^{\mathrm{LOG}}$ where $\lambda_{theory} = 2\sqrt{\log p/n}$ . . . . . 37
2.11 For the three estimators $\left(\hat{\boldsymbol{\Sigma}}^{\mathrm{mGL}}, \hat{\boldsymbol{\Sigma}}^{\mathrm{GL}}, \hat{\boldsymbol{\Sigma}}^{\mathrm{LOG}}\right)$, $MSE(\lambda_{best})$ as a function of $K$ under the moving average pattern (Left) and the stair pattern (Right) where $\lambda_{best} = \arg\min_{\lambda\in\Lambda} MSE(\lambda)$. . . . . 37

3.1 A tree that relates adjectives on its leaves . . . . . 45
3.2 (Left) An example of $\boldsymbol{\beta} \in \mathbb{R}^5$ and $\mathcal{T}$ that relates the corresponding five features. By (3.4), we have $\beta_i = \gamma_i + \gamma_6 + \gamma_8$ for $i = 1, 2, 3$ and $\beta_j = \gamma_j + \gamma_7 + \gamma_8$ for $j = 4, 5$. (Right) By zeroing out the $\gamma_i$'s in the gray nodes, we aggregate $\beta$ into two groups indicated by the dashed contours: $\beta_1 = \beta_2 = \beta_3 = \gamma_6 + \gamma_8$ and $\beta_4 = \beta_5 = \gamma_8$. Counts data are aggregated for features sharing the same coefficient: $\boldsymbol{X\beta} = (\boldsymbol{X}_1 + \boldsymbol{X}_2 + \boldsymbol{X}_3)\beta_1 + (\boldsymbol{X}_4 + \boldsymbol{X}_5)\beta_4$. . . . . 52
3.3 In the above tree, $B^* = \{u_1, u_2, u_3, u_4, y_5\}$ has its nodes labeled with black circles. . . . . 56
3.4 (Left and Middle) two scenarios for varying $k$: $\min_\Lambda \|\hat{\boldsymbol{\beta}}(\Lambda) - \boldsymbol{\beta}^*\|_2^2/p$ versus $k$ for $(n, p, s) = (500, 100, 0)$ and $(n, p, s) = (100, 200, 0.2)$. (Right) degredation of our method with distorted trees: $\min_\Lambda \|\boldsymbol{X}\hat{\boldsymbol{\beta}}(\Lambda) - \boldsymbol{X\beta}^*\|_2^2/n$ versus $\tau$ for $(n, p, s, k) = (100, 200, 0.2, 10)$. . . . . 60
3.5 (Left) distribution of TripAdvisor ratings. (Right) only 414 adjectives appear in more than 1% of reviews; the histogram gives the distribution of usage-percentages for those adjectives appearing in fewer than 1% of reviews. . . . . 62
3.6 Tree $\mathcal{T}$ over 2,397 adjectives: the left subtree is for adjectives with negative sentiment and the right subtree is for adjectives with positive sentiment. . . . . 62
3.7 A comparison between our method and four other methods . . . 63
3.8 Trees for 2,397 adjectives on the leaves with branches colored based on $\hat{\boldsymbol{\beta}}$ estimated with the lasso (Top) and our method (Bottom), respectively. Red branch, blue branch and gray branch correspond to negative, positive and zero $\hat{\beta}_j$, respectively. Darker color indicates larger magnitude of $\hat{\beta}_j$ and lighter color indicates smaller magnitude of $\hat{\beta}_j$. . . . . 65

3.9 $\{|\hat{\beta}_j|\}$ versus term density (on log scale) for adjectives selected by our method (black circles) and the lasso (red triangles) in the $n = 1,700$ and $p = 2,397$ case. . . . . 65

4.1 (Top) phylogenetic tree with $\gamma_u$ assigned to node $u$ and a taxonomic level labeled for every depth. When aggregating OTU counts to genus level (shaded in gray), the OTUs naturally separate into four subtrees with leaves colored accordingly. (Bottom) A more flexible aggregation pattern induced by zeroing out $\gamma_u$'s in the crossed nodes. The roots of aggregated subtrees are shaded in gray and the corresponding OTUs are colored accordingly on the leaves. In both examples, $\beta_j$'s from the same aggregation share equal values. . . . . 76

4.2 (Left) Distribution of OTU densities for 481 OTUs in 100 samples used in simulation. (Right) Generated $\tilde{\boldsymbol{\beta}}^*$ elements for the 113 true aggregations. . . . . 79

4.3 Phylogenetic tree built upon taxonomy matrix for the 481 OTUs used in simulation. OTU labels are on the leaves. There are seven taxonomic levels upon OTU: species, genus, family, order, class, phylum and kingdom; each level corresponds to a depth in the tree. The 481 OTUs are either kept at OTU level by themselves or aggregated up to the class level. Subtrees are colored accordingly to illustrate the true aggregations. . . . . 80

4.4 (Top Left) $\min_\Lambda \|\hat{\boldsymbol{\beta}}(\Lambda) - \boldsymbol{\beta}^*\|_2^2/p$ and (Top Right) $\min_\Lambda \|\hat{\boldsymbol{y}}(\Lambda) - \log(\boldsymbol{Z})\boldsymbol{\beta}^*\|_2^2/n$ versus varying `snr`. (Bottom) Fixing `snr` = 10, (Bottom Left) $\min_\Lambda \|\hat{\boldsymbol{\beta}}(\Lambda) - \boldsymbol{\beta}^*\|_2^2/p$ and (Bottom Right) $\min_\Lambda \|\hat{\boldsymbol{y}}(\Lambda) - \log(\boldsymbol{Z})\boldsymbol{\beta}^*\|_2^2/n$ versus increasing proportions of OTUs missing species, genus and family labels. . . . . 83

4.5 In the gut microbiome data, distribution of microbe densities at OTU level (Left) and at genus level (Middle) (after filtering at 5% threshold of density). Our method at its best performance aggregates 761 OTUs into 152 taxa with non-zero coefficients. The right panel overlays the densities of the 761 OTUs (in red) and that of the 152 taxa (in blue). . . . . 86

4.6 (Left) Test set MSE versus model size at every $(\lambda_1, \lambda_2)$ of our method, where model size for (4.4) is the number of resulting aggregations with non-zero coefficients. At OTU level (Middle) and at genus level (Right), Test set MSE versus model size at every $\lambda$ of (4.2), where model size for (4.2) is the number of selected taxa. Red and blue points correspond to the tuning parameters selected by CV and that achieves the lowest test error, respectively. 87

4.7 $\{|\hat{\beta}_j|\}$ versus density (on log scale) for OTUs selected by our method (black circles) and (4.2) at OTU level (red triangles), for genera selected by (4.2) at genus level (green squares). . . . . 88

A.1 For the three estimators ($\hat{\mathbf{\Sigma}}^{\text{mGL}}, \hat{\mathbf{\Sigma}}^{\text{GL}}, \hat{\mathbf{\Sigma}}^{\text{LOG}}$) in moving-average pattern, probability of their estimates being PSD at $\lambda_{best}$. . . . . 119
A.2 For the three estimators ($\hat{\mathbf{\Sigma}}^{\text{LOG}}, \hat{\mathbf{\Sigma}}^{\text{mGL}}, \hat{\mathbf{\Sigma}}^{\text{GL}}$) in moving-average pattern, minimum eigenvalues of 50 samples at $\lambda_{best}$. . . . . 120

CHAPTER 1

# INTRODUCTION

Structural patterns in parameters, including *structured sparsity patterns* and *structured equality patterns*, are desired in many statistics problems for various reasons. For example, in an interaction model one zeros out an interaction effect if any of its main effects are not selected, so that the estimated model is more interpretable. Such structured sparsity pattern is rooted from prior knowledge of the problem. Meanwhile, the decision of enforcing structured equality among similar features is often made out of practical reason: modeling highly sparse data is hard because of the lack of variability. To overcome the difficulty of sparsity in the data itself, appropriate aggregations of relevant features is a natural choice, which can be achieved by setting groups of parameters equal in an additive model setting. We focus on the following two challenging cases in getting structural patterns in parameters, for which existing studies are inadequate:

1. Hierarchical sparse modeling (HSM) in which one group of parameters is set to zero whenever another is set to zero and such hierarchical sparsity pattern is encoded in a directed acyclic graph (DAG) (Chapter 2); and
2. Highly sparse count data which arise when features record frequency of events (or the number of times certain properties hold) and a large fraction of the events are rare (Chapter 3 and Chapter 4).

The methods under consideration share two similarities. First, they both use sparsity-inducing convex regularization procedures to induce respective structural pattern. In HSM, we compare two frameworks, the group lasso (GL) and latent overlapping group lasso (LOG), for getting hierarchical sparsity patterns

in parameters. For modeling highly sparse count data, we convert a feature aggregation problem to a sparse modeling problem with the proposed tree-based parametrization and $\ell_1$ regularization. Second, in both scenarios we require side information in the form of either a tree or a DAG that encodes relations among the features. In HSM, one forms a DAG with parameters embedded in its nodes to encode the desired hierarchical sparsity relations among the parameters. When it comes to rare count features, the tree is grown upon the parameters on the leaves, and branches are merged based on their similarity.

In Chapter 2, we provide a side-by-side comparison of GL and LOG in HSM in terms of their statistical properties and computational efficiency. We call special attention to GL's more aggressive shrinkage of parameters deep in the hierarchy, a property not shared by LOG. In terms of computation, we introduce a finite-step algorithm that exactly solves the proximal operator of LOG for a certain simple HSM structure; we later exploit this to develop a novel path-based block coordinate descent scheme for general HSM structures. Both algorithms greatly improve the computational performance of LOG. Finally, we compare the two methods in the context of covariance estimation, where we introduce a new sparsely-banded estimator using LOG, which we show achieves the statistical advantages of an existing GL-based method but is simpler to express and more efficient to compute.

In Chapter 3, we describe the difficulty in modern prediction problems when many features are counts of rarely occurring events, and the prevalence of such "rare features" in diverse areas, ranging from natural language processing (e.g., rare words) to biology (e.g., rare species). We show, both theoretically and empirically, that not explicitly accounting for the rareness of features can greatly

reduce the effectiveness of an analysis. We next propose a framework for aggregating rare features into denser features in a flexible manner that creates better predictors of the response. Our strategy leverages side information in the form of a tree that encodes feature similarity, and is formulated as a solution to a convex optimization problem. We apply our method to data from TripAdvisor, in which we predict the numerical rating of a hotel based on the text of the associated review. Our method achieves high accuracy by making effective use of rare words; by contrast, the lasso is unable to identify highly predictive words if they are too rare.

In Chapter 4, we apply the proposed aggregation framework from Chapter 3 on compositional data in microbiome analysis that measures relative abundance among the observed microbial species. We integrate the tree-based framework into a linear log-contrast model for which log-transformed proportions are treated as features, subject to a zero-sum constraint on the regression coefficients. A phylogenetic tree that joins microbial species based on their taxonomic similarities is used to guide the feature aggregation. We apply our tree-guided log-contrast model to data from the American Gut project, in which we predict one's BMI based on both microbiome and non-microbiome features. Our method achieves better prediction accuracy than the conventional log-contrast model that requires aggregation at genus or higher levels and filtering, and yields biologically more interpretable results.

CHAPTER 2

# A CHOICE OF TWO REGULARIZERS IN HIERARCHICAL SPARSE MODELING

*Portions of this chapter were published in Yan and Bien (2017).*

## 2.1 Introduction

Convex regularizers for sparse modeling are ubiquitous in the statistics and machine learning literatures. Regularizers such as the *lasso* (Tibshirani, 1996) and the *group lasso* (Turlach et al., 2005; Yuan and Lin, 2006) are commonly-used tools for seemlessly integrating model selection into statistical procedures, thereby extending these methods' reach to high-dimensional settings in which the number of parameters greatly exceeds the sample size. In contrast to the lasso, which seeks sparsity with no *a priori* pattern, the group lasso regularizer allows pre-defined groups of variables to be set to zero simultaneously, giving rise to the so-called *structured sparsity* literature in which certain patterns of zeros are sought (Bach et al., 2012). The focus of this chapter is on a particular kind of structured sparsity that arises in many statistics problems, which we will call *hierarchical sparse modeling* (HSM). Given a vector $\beta \in \mathbb{R}^p$ of parameters and a known collection of non-empty, disjoint sets $s_1, \cdots, s_N \subseteq \{1, \ldots, p\}$, HSM focuses on situations in which we wish to set groups of variables to zero while ensuring that

$$\beta_{s_i} = 0 \implies \beta_{s_j} = 0$$

for certain ordered pairs of groups $(s_i, s_j)$. More specifically, in HSM one forms a directed acyclic graph (DAG) over $\{s_1, \cdots, s_N\}$ to encode the desired hierarchical sparsity relations (one requires the above to hold if $s_i$ is an ancestor of $s_j$ in the

DAG). HSM appears in many applications in statistics, including interactions (Yuan et al., 2009; Zhao et al., 2009; Radchenko and James, 2010; Schmidt and Murphy, 2010; Choi et al., 2010; Jenatton et al., 2010; Bien et al., 2013; Lim and Hastie, 2015; She et al., 0; Haris et al., 2016), covariance matrix estimation (Levina et al., 2008; Rothman et al., 2010; Bien et al., 2016), additive models (Lou et al., 2016; Chouldechova and Hastie, 2015), time series models (Nicholson et al., 2014), and multiple kernel learning (Bach, 2008). We note that *hierarchical sparse coding* is a common special case of HSM in which the DAG is a forest of trees (Zhao et al., 2009; Jenatton et al., 2011b). For example, in a two-way interaction model of the form

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_{12} X_1 X_2 + \beta_{13} X_1 X_3 + \beta_{23} X_2 X_3 + \epsilon,$$

one can express the principle of marginality (Nelder, 1977) as that $\beta_j$ and $\beta_k$ are parents of $\beta_{jk}$ (each node of the DAG contains a single element, i.e., $|s_i| = 1$ for all $i$). The DAG, which is not a tree, is depicted in Figure 2.1. A simpler DAG structure arises in banded covariance estimation, in which a $p \times p$ matrix $\Sigma$'s sparsity pattern can be described by having the elements of each subdiagonal set to zero only if those farther from the main diagonal than it are also all set to zero (in this situation, the DAG is simply a path as depicted in Figure 2.3 with $D = p - 1$). We will discuss banded covariance estimation in greater detail in Section 2.5.

There are two primary convex regularizers used for structured sparsity: the *group lasso* (GL) and *latent overlapping group lasso* (LOG) (Jacob et al., 2009). The sparsity patterns attained by these regularizers are in general different in nature, and so the regularizers typically arise in complementary situations. Given a set of groups of parameters $\mathcal{G}$, GL *sets to zero a union of groups* that is a subset of $\mathcal{G}$. The GL penalty is defined as a weighted sum of $\ell_2$ norms over groups of

parameters as defined in $\mathcal{G}$:

$$\Omega_{\text{GL}}^{\mathcal{G}}(\beta; w) = \sum_{g \in \mathcal{G}} w_g \|\beta_g\|_2. \tag{2.1}$$

Here, $w_g$ are positive scalars that control the relative strength of the terms within the GL penalty.

Jacob et al. (2009) observe that when the groups in $\mathcal{G}$ overlap, the induced support from GL may not be a union of groups since the complement of a union of groups is not necessarily a union of groups. In this sense, the group lasso as defined in (2.1) should not be used in situations in which one wishes a subset of (overlapping) groups to remain nonzero. The authors propose LOG as a solution to this problem. Rather than apply the $\ell_1/\ell_2$ norm directly on the parameter vector $\beta$, LOG forms the parameters as a sum of GL-penalized latent variables, which is each supported by a group $g$:

$$\Omega_{\text{LOG}}^{\mathcal{G}}(\beta; w) = \inf_{\{v^{(g)} \in \mathbb{R}^p\}_{g \in \mathcal{G}}} \left\{ \sum_{g \in \mathcal{G}} w_g \|v^{(g)}\|_2 \quad \text{s.t.} \quad \sum_{g \in \mathcal{G}} v^{(g)} = \beta \text{ and } v_{g^c}^{(g)} = 0 \text{ for } g \in \mathcal{G} \right\}. \tag{2.2}$$

In LOG, a subset of the latent variables is set to zero. Since $\beta$ is formed as a sum of these latent variables, the parameters in a group $g$ are selected as long as the corresponding latent variable $v^{(g)}$ is nonzero. As a result, the LOG penalty *leaves nonzero a union of groups*.

Although GL and LOG induce different sparsity patterns in general, we show in Section 2.2 that in the special case of HSM, either regularizer (with an appropriately chosen group structure) can be used to accomplish the HSM structure. From a methodological statistician's standpoint, this observation leads to ambiguity as to which regularizer one should use for HSM. Indeed, a survey of the HSM literature reveals that researchers have been using both

Table 2.1: Applications of GL and LOG in HSM

| Problem | Group Lasso (GL) | Latent Overlapping GL (LOG) |
|---|---|---|
| Hierarchical Interactions | `CAP`, Zhao et al. (2009)<br>`VANISH`, Radchenko and James (2010)<br>Schmidt and Murphy (2010)<br>`hiernet`, Bien et al. (2013)<br>`GRESH`, She et al. (0)<br>`FAMILY`, Haris et al. (2016) | `glinternet`,<br>Lim and Hastie (2015) |
| Banded Covariance Matrix | `hierband`, Bien et al. (2016) | Section 2.5 of this chapter |
| Generalized Partially Linear Additive Models | `SPLAM`, Lou et al. (2016) | `GAMSel`,<br>Chouldechova and Hastie (2015) |
| Times Series | `HVAR`, Nicholson et al. (2014) | — |
| Hierarchical Multiple Kernel Learning | `HKL`, Bach (2008) | — |

frameworks with no discussion of the seemingly arbitrary choice about whether to use GL or LOG. Table 2.1 arranges methods developed across five statistical domains according to which regularizer was used. One observes that LOG is the less commonly employed regularizer in HSM problems. The objective of this chapter is to compare the GL and LOG approaches in the context of HSM. While the class of sparsity patterns obtainable is the same for the two regularizers, we show in Section 2.2.3 that the nature of the shrinkage is different even for the simplest nontrivial HSM problem. The main contributions of our investigation into these two regularizers are summarized below:

- In Section 2.3, we show that the GL penalty as defined in (2.1) tends to apply a greater amount of shrinkage to parameters embedded deep in the DAG whereas LOG does not. In certain situations where this more aggressive shrinkage is not desired, a more complicated weighting scheme can be adopted (as was done in Jenatton et al. 2011a; Bien et al. 2016). This weighting scheme, which makes computation and theory more involved, appears to be necessary to match the statistical performance of LOG.

- In Section 2.4, we focus on computational aspects. It was shown in Jenatton et al. (2011b) that when the DAG is a tree, the proximal operator of GL could be solved exactly in a finite number of operations. While there is no known corresponding algorithm for LOG, in the special case that the DAG is a path graph (or forest of path graphs), we derive such an algorithm. We then leverage this result to introduce a novel path-based block coordinate descent (BCD) scheme for the case of a general DAG that is more efficient than the standard BCD algorithm.
- In Section 2.5, as a case study, we demonstrate how the LOG framework can be used instead of GL for the problem of estimating a banded covariance matrix. We use banded covariance matrix estimation as a primary basis to compare the statistical performance between the GL and LOG frameworks. We prove that this estimator attains the same bandwidth recovery properties and convergence rate as the "convex banding" estimator of Bien et al. (2016), which had to rely on a complicated weighting scheme. Furthermore, we find that it attains similar empirical performance.

**Notation:** We use $\|\beta\|_2$ and $\|\mathbf{\Sigma}\|_F$ for the $\ell_2$ norm of a vector $\beta \in \mathbb{R}^p$ and the Frobenius norm of a matrix $\mathbf{\Sigma} \in \mathbb{R}^{p\times p}$, respectively. The support of $\beta$ is denoted $\text{supp}(\beta) \subseteq \{1, \ldots, p\}$, which is the set of indices of nonzero elements in $\beta$. For $\beta$, a group of parameters is a subset $g \subseteq \{1, \ldots, p\}$. We use $\mathcal{G}$ to denote the set of groups. The weight vector $w$, of the same size as $\mathcal{G}$, has positive elements. For a group $g \subseteq \{1, \ldots, p\}$, $\beta_g \in \mathbb{R}^p$ has the same entries as $\beta$ for indices in $g$ and is 0 for all other indices, whereas $\beta_{|g} \in \mathbb{R}^{|g|}$ is a subset of $\beta$ for indices in $g$. For a matrix $\mathbf{X} \in \mathbb{R}^{n\times p}$ and a subset $g \subseteq \{1, \ldots, p\}$, $\mathbf{X}_{|g} \in \mathbb{R}^{n\times|g|}$ has the same columns as $\mathbf{X}$ for column indices in $g$. In Section 2.5, given a subset of a matrix indices

$g \subseteq \{1, \ldots, p\}^2$ of a matrix $\mathbf{\Sigma}$, let $\mathbf{\Sigma}_g \in \mathbb{R}^{p \times p}$ be a matrix whose entries are the same as $\mathbf{\Sigma}$ for the indices in $g$, and are 0 for other indices. Let $(\cdot)_+ = \max\{\cdot, 0\}$ denote the positive part and $S(\cdot, \cdot)$ and $S_G(\cdot, \cdot)$ the elementwise and groupwise soft-thresholding operators, respectively:

$$[S(y, \mu)]_i = y_i \left(1 - \frac{\mu}{|y_i|}\right)_+ \quad \text{and} \quad S_G(y, \mu) = y \left(1 - \frac{\mu}{\|y\|}\right)_+,$$

where $\|\cdot\|$ denotes $\|\cdot\|_2$ or $\|\cdot\|_F$, depending on whether $y$ is a vector or a matrix.

## 2.2 Hierarchical Sparse Modeling: Two Frameworks

Let $s_1, \cdots, s_N \subseteq \{1, \ldots, p\}$ be a collection of nonempty, disjoint sets of indices and let $\mathcal{D}$ be a DAG with vertex set $\{s_1, \cdots, s_N\}$. In specifying a DAG, the notions of *ancestor* and *descendant* are well-defined. In particular, we let descendants$(\mathcal{D}; s_i)$ denote the set of all $s_j$ for which there exists a path from $s_i$ to $s_j$ in $\mathcal{D}$ and we likewise let ancestors$(\mathcal{D}; s_j)$ denote the set of all $s_i$ for which there exists a path from $s_i$ to $s_j$. Note that we let a node itself be in both its ancestor group and its descendant group. To better illustrate the constructions of *ancestor* and *descendant*, we use a two-way interaction model with three predictors as an example. The corresponding DAG for the interaction model is shown in Figure 2.1. To be specific, for each main effect $\beta_j$, the two interaction effects resulted from $\beta_j$ and another main effect $\beta_k$ are considered as descendancts of $\beta_j$. Conversely, for the interaction effect $\beta_{jk}$, its two parent main effects, $\beta_j$ and $\beta_k$, are its ancestors.

The goal of HSM is to attain sparsity patterns for which

$$\beta_{s_i} = 0 \quad \Rightarrow \quad \beta_{s_j} = 0 \quad \text{for all } s_j \in \text{descendants}(\mathcal{D}; s_i). \tag{2.3}$$

In the context of our interaction model example, (2.3) enforces the selection that

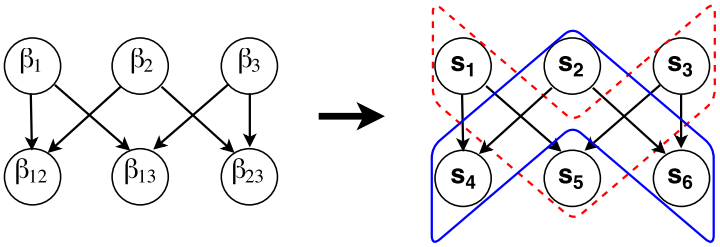


Figure 2.1: (Left) A DAG $\mathcal{D}$ for a two-way interaction model with three predictors. In HSM, the DAG $\mathcal{D}$ encodes the sparsity structure: a node's parameters must be set to zero if it has a parent with zeroed parameters. (Right) The same $\mathcal{D}$ specified using our notation: each node contains only one element and the correspondence between $s_i$ and $\beta_j$ is as shown. In red dashed contour, ancestors$(\mathcal{D}; s_5) = \{s_1, s_3, s_5\}$ include both main effects, $\beta_1$ and $\beta_3$, in the ancestor group of the interaction effect $\beta_{13}$. In blue solid contour, descendants$(\mathcal{D}; s_2) = \{s_2, s_4, s_6\}$ contains both interaction effects involving main effect $\beta_2$.

all the resulting interaction effects are discarded if the main effect is not selected.

We can equivalently express (2.3) as

$$\beta_{s_j} \neq 0 \quad \Rightarrow \quad \beta_{s_i} \neq 0 \quad \text{for all } s_i \in \text{ancestors}(\mathcal{D}; s_j). \tag{2.4}$$

In interaction modeling, this tells us that all its parent main effects need to be selected if an interaction effect is selected. Given (2.3) and (2.4) are functionally equivalent statements, we show in Sections 2.2.1 and 2.2.2 how GL and LOG are based on (2.3) and (2.4), respectively. While their sparsity patterns are equivalent, we show in Section 2.2.3 that the two approaches lead to different solutions.

### 2.2.1 The Group Lasso Approach

To induce the hierarchical sparsity of (2.3), Zhao et al. (2009), Jenatton et al. (2011b) and many others use the GL regularizer (2.1) with group structure $\mathcal{G}$ chosen to be

$$d(\mathcal{D}) := \{\text{descendants}(\mathcal{D}; s_i) : i = 1, \ldots, N\}. \tag{2.5}$$

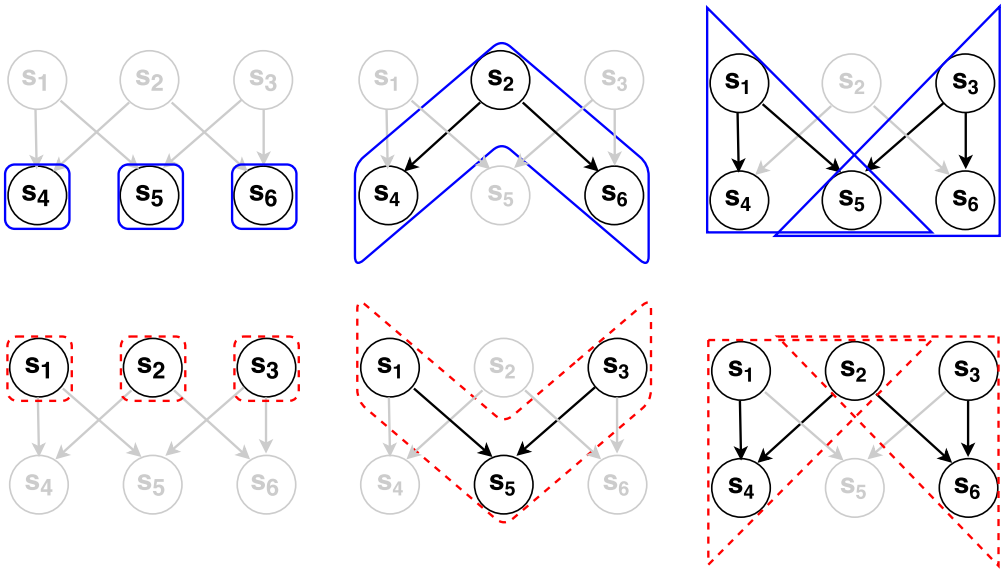


Figure 2.2: For the same DAG as in Figure 2.1, an illustration of group structures $\mathcal{G} = d(\mathcal{D})$ and $\mathcal{G} = a(\mathcal{D})$ induced for GL and LOG, respectively. (Top) The group structure $d(\mathcal{D})$ for GL is shown in solid contours: $d(\mathcal{D}) = \{s_4, s_5, s_6, s_2 \cup s_4 \cup s_6, s_1 \cup s_4 \cup s_5, s_3 \cup s_5 \cup s_6\}$. Each group of $d(\mathcal{D})$ can be thought of as a set of the efffect itself and all the relevant interaction effects. (Bottom) The group structure $a(\mathcal{D})$ for LOG is shown in dashed contours: $a(\mathcal{D}) = \{s_1, s_2, s_3, s_1 \cup s_3 \cup s_5, s_1 \cup s_2 \cup s_4, s_2 \cup s_3 \cup s_6\}$. Each group of $a(\mathcal{D})$ can be described as a set of the effect itself and all the relevant main effects.

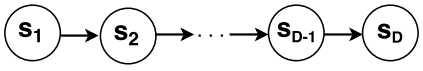


Figure 2.3: Directed Path Graph with $D$ Nodes

The top panels of Figure 2.2 gives an example of $d(\mathcal{D})$ for a DAG associated with a two-way interaction model with three predictors. There is a group corresponding to each node $s_i$, and this group contains all the parameters in $s_i$ and in its descendant nodes. Recalling that GL sets to zero a union of groups, we see that $\Omega_{\text{GL}}^{d(\mathcal{D})}$ achieves (2.3). As shown in the top panels of Figure 2.2, each main effect is grouped with its descendant interaction effects, whereas each interaction effect is grouped by itself. It is possible for an interaction effect to be zeroed out while keeping its parent main effects significant. However, whenever the

main effect is zeroed out which only occurs when the whole group (including interaction effects) is not selected, all the descendant interaction effects must be zeroed out as well. We choose a convex smooth loss function $F$ depending on the statistical context (a common choice is the negative log-likelihood) and then solve

$$\min_{\beta \in \mathbb{R}^p} \left\{ F(\beta) + \lambda \Omega_{\mathrm{GL}}^{d(\mathcal{D})}(\beta; w) \right\}. \tag{2.6}$$

Here, $\lambda \geq 0$ is a regularization parameter that controls the sparsity level of $\beta$.

### 2.2.2 The Latent Overlapping Group Lasso Approach

The LOG penalty (2.2) of Jacob et al. (2009) can be used for HSM taking the perspective of (2.4). We choose $\mathcal{G}$ to be

$$a(\mathcal{D}) := \left\{ \text{ancestors}(\mathcal{D}; s_j) : j = 1, \ldots, N \right\}. \tag{2.7}$$

For each node $s_j$ in $\mathcal{D}$, there is a group containing all parameters that are contained in $s_j$ or its ancestors. The bottom panels of Figure 2.2 shows $a(\mathcal{D})$ for the same DAG as on the top. As observed in Jacob et al. (2009), LOG leaves a union of groups nonzero, thus we see that (2.4) is accomplished by $\Omega_{\mathrm{LOG}}^{a(\mathcal{D})}$. In our interaction model example, as shown in the bottom panels of Figure 2.2, each interaction effect is grouped with both parent main effects, whereas each main effect is grouped by iteself separately. This group structure guarantees (2.4) since both main effects will be recovered as nonzero if we have a nonzero interaction effect, given they are in the same group. We are thus faced with a choice of whether to use an estimator defined based on solving (2.6) versus one based on solving

$$\min_{\beta \in \mathbb{R}^p} \left\{ F(\beta) + \lambda \Omega_{\mathrm{LOG}}^{a(\mathcal{D})}(\beta; w) \right\}. \tag{2.8}$$

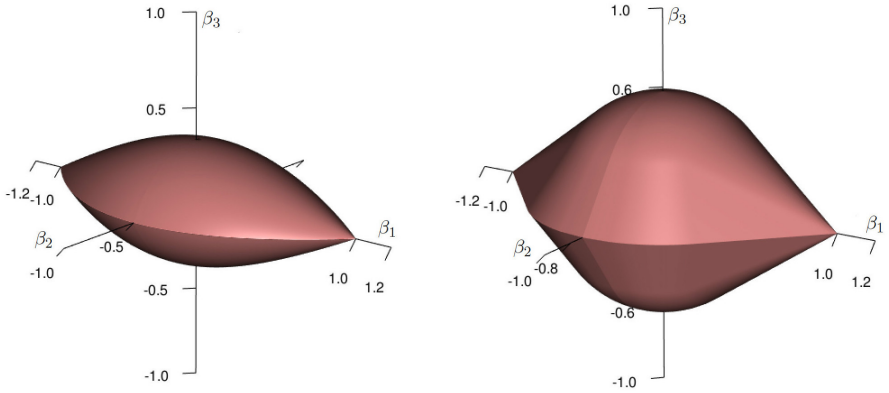


Figure 2.4: For $\beta \in \mathbb{R}^3$ and the DAG $\{1\} \to \{2\} \to \{3\}$, (Left) the unit ball of $\Omega_{\text{GL}}^{d(\mathcal{D})}(\beta; w)$ where $d(\mathcal{D}) = \{\{1,2,3\},\{2,3\},\{3\}\}$ and $w = (1,1,1)$ and (Right) the unit ball of $\Omega_{\text{LOG}}^{a(\mathcal{D})}(\beta; w)$ where $a(\mathcal{D}) = \{\{1\},\{1,2\},\{1,2,3\}\}$ and $w = (1, \sqrt{2}, \sqrt{3})$.

### 2.2.3 Are These Two Approaches Different?

In Sections 2.2.1 and 2.2.2 we describe two frameworks that lead to the same set of sparsity patterns. This equivalence can be shown geometrically in the simple case in which $p = 3$, $s_i = \{i\}$ for $i = 1, 2, 3$, and $\mathcal{D}$ is the path graph $s_1 \to s_2 \to s_3$. Figure 2.4 depicts the unit ball of the induced GL and LOG penalties introduced in the previous sections. We observe that both balls have their nondifferentiable points lying in the plane defined by $\beta_3 = 0$. Furthermore, both unit balls have "poles" on the axis defined by $\beta_2 = \beta_3 = 0$. Given that both penalties lead to the same set of supports, it is natural to ask if these two regularizers are in fact identical for an appropriately chosen set of weights. We consider the simplest nontrivial HSM: let $p = 2$, $s_1 = \{1\}$ and $s_2 = \{2\}$, and take $\mathcal{D}$ to be a single edge connecting singleton sets: $s_1 \to s_2$. The following lemma establishes that these two penalties are different even in this simplest of situations.

**Lemma 1.** *Take $\mathcal{D}$ to be $\{1\} \to \{2\}$ and fix $w' = (1,1)$. There does not exist $w \in \mathbb{R}^{+2}$ such that*

$$\Omega_{\text{GL}}^{d(\mathcal{D})}(\beta; w) = \Omega_{\text{LOG}}^{a(\mathcal{D})}(\beta; w') \quad \forall \beta \in \mathbb{R}^2.$$

*Proof.* See Appendix A.1. □

Moreover, we can compare the proximal operators of the two penalties, which correspond to (2.6) and (2.8) with $F(\beta) = \frac{1}{2}\|y - \beta\|_2^2$:

$$\text{Prox}_{\text{GL}}^{d(\mathcal{D})}(y; \lambda, w) := \underset{\beta \in \mathbb{R}^p}{\arg\min} \left\{ \frac{1}{2} \|y - \beta\|_2^2 + \lambda \Omega_{\text{GL}}^{d(\mathcal{D})}(\beta; w) \right\}, \tag{2.9}$$

$$\text{Prox}_{\text{LOG}}^{a(\mathcal{D})}(y; \lambda, w) := \underset{\beta \in \mathbb{R}^p}{\arg\min} \left\{ \frac{1}{2} \|y - \beta\|_2^2 + \lambda \Omega_{\text{LOG}}^{a(\mathcal{D})}(\beta; w) \right\}. \tag{2.10}$$

The use of equality in the above definition is justified by observing that $F$ is strongly convex and therefore the arg min is a single point. The path graph structure of the simplest HSM example allows us to express both proximal operators in closed form, which allows us to see plainly how they differ. Let $\hat{\beta}^{\text{GL}}$ and $\hat{\beta}^{\text{LOG}}$ denote the solution to the respective proximal operators defined in (2.9) and (2.10).

**Lemma 2.** *Taking $\mathcal{D}$ to be $\{1\} \to \{2\}$, $\hat{\beta}^{\text{GL}}$ and $\hat{\beta}^{\text{LOG}}$ can be written in closed form:*

$$\hat{\beta}^{\text{GL}} = S_G\left(\begin{pmatrix} y_1 \\ S\ (y_2, \lambda w_2) \end{pmatrix}, \lambda w_1\right)$$

$$\hat{\beta}^{\text{LOG}} = \begin{cases} S_G\,(y, \lambda w_2) & \text{if } |y_2| \geq \frac{\sqrt{w_2^2 - w_1^2}}{w_1} |y_1| \\ \begin{pmatrix} S\,(y_1, \lambda w_1) \\ S\,(y_2, \lambda \sqrt{w_2^2 - w_1^2}) \end{pmatrix} & \text{otherwise} \end{cases}$$

*with $w_1$ and $w_2$ in GL being applied on the group $\{1, 2\}$ and $\{2\}$, respectively, and $w_1$ and $w_2$ in LOG being applied on the group $\{1\}$ and $\{1, 2\}$, respectively.*

*Proof.* This result follows by applying Algorithms 1 and 3 in Section 2.4. □

We see that $\hat{\beta}_2^{\text{GL}}$ has two "chances" to be set to zero: first, through the elementwise soft thresholding of $y_2$ and, second, through the groupwise soft-thresholding of $(y_1, S(y_2, \lambda w_2))$. By contrast, for $\hat{\beta}_2^{\text{LOG}}$, the shrinkage is applied only once (though whether it is an elementwise or groupwise soft-thresholding depends on the relative size of $|y_1|$ and $|y_2|$). This example establishes that these two regularizers are in fact different, so we proceed to investigate the nature and implications of this difference.

## 2.3 Differential Shrinkage of GL

In this section, we call attention to a property of the GL shrinkage that is not shared by LOG: namely, that $\Omega_{\text{GL}}^{d(\mathcal{D})}$ shrinks parameters embedded in nodes deep in the DAG $\mathcal{D}$ more agressively than those that are in less deep nodes in the DAG. This "over-penalization" phenomenon has been observed previously (Jenatton et al., 2011a; Bach et al., 2012; Bien et al., 2016) in overlapping group lasso settings, but it does not appear to be widely appreciated. A simple explanation for this phenomenon is that the vector $\beta_{s_j}$ appears within $\Omega_{\text{GL}}^{d(\mathcal{D})}$ in $|\text{ancestors}(\mathcal{D}; s_j)|$ terms, a number that can vary greatly among different $s_j$. In Section 2.4, we will see that the amount of shrinkage of $\beta_{s_j}$ grows with the number of groups its indices $s_j$ belong to. For example, for the path graph $\mathcal{D}$ shown in Figure 2.3, $\beta_{s_1}$ appears in only a single groupwise soft-thresholding whereas $\beta_{s_D}$ is soft-thresholded $D$ times. The uneven distribution of shrinkage over the support in GL is a nonnegligible phenomenon. By contrast, we will show that $\Omega_{\text{LOG}}^{a(\mathcal{D})}$ applies a comparable amount of shrinkage at all depths of $\mathcal{D}$.

In order to more directly study the difference of the shrinking mechanisms

in GL and LOG, we will compare the solutions to (2.9) and (2.10) for the directed path graph in Figure 2.3 in the case that there is one parameter per node, i.e., $s_i = \{i\}$ for $i = 1, \ldots, D$. For simplicity, we consider $y \sim N_D(\beta^*, \sigma^2 I_D)$ where $\beta^*$ is an unknown mean vector. The group structure $d(\mathcal{D})$ for GL for this DAG consists of groups of the form $\{i, \cdots, D\}$ for $i = 1, \ldots, D$. For $\lambda \geq 0$, we compute

$$\hat{\beta}^{\text{GL}} = \text{Prox}_{\text{GL}}^{d(\mathcal{D})}(y; \lambda, \{w_i = 1\}). \tag{2.11}$$

Likewise, the group structure $a(\mathcal{D})$ for LOG consists of groups of the form $\{1, \cdots, i\}$ for $i = 1, \ldots, D$, and we compute

$$\hat{\beta}^{\text{LOG}} = \text{Prox}_{\text{LOG}}^{a(\mathcal{D})}(y; \lambda, \{w_i = \sqrt{i}\}). \tag{2.12}$$

The following two propositions emphasize the difference between the penalties in terms of the "over-penalization" phenomenon.

**Proposition 1.** *Let $\beta_d^* = 1_{\{d \leq K^*\}}$ for $K^* < D$. For $\hat{\beta}^{\text{GL}}$ in (2.11), if we choose $\lambda > \bar{\lambda} := 2\sigma\sqrt{\log D}$, then with probability at least $1 - 2/D$,*

*(a) $supp(\hat{\beta}^{\text{GL}}) \subseteq supp(\beta^*)$*

*(b) For $1 \leq d \leq d + h \leq K^*$ and $\hat{\beta}_d^{\text{GL}} \neq 0$,*

$$\frac{\left|\hat{\beta}_{d+h}^{\text{GL}}\right|}{\left|\hat{\beta}_d^{\text{GL}}\right|} \leq \frac{|y_{d+h}|}{|y_d|} \exp\left(-\frac{\lambda h}{\sqrt{\sum_{m=d+1}^{K^*} y_m^2}}\right). \tag{2.13}$$

*Proof.* See Appendix A.2.1. □

Equation (2.13) shows that the difference in the amount of shrinkage applied to two elements in $\mathcal{D}$ increases at least exponentially with the distance $h$ between them. In particular, Proposition 1 illustrates the differential shrinkage of GL: parameters embedded in nodes deep in the DAG are shrunken more aggressively

than those that are in less deep nodes. Indeed, we can see this exponential decaying pattern empirically in two examples shown in the left panels of Figure 2.5 and Figure 2.6. The next proposition shows that LOG by contrast applies a uniform shrinkage across all elements.

**Proposition 2.** *For the same $\beta^*$ as in Proposition 1 and $\hat{\beta}^{\mathrm{LOG}}$ in* (2.12)*, assuming $D > 1$ and $\bar{\lambda} := 2\sigma\sqrt{\log D} < 1$, if we choose*

$$\bar{\lambda} < \lambda \leq (1-\delta)(1-\bar{\lambda}),$$

*for $\delta \in (0,1)$ then with probability at least $1 - 2/D$,*

*(a)* $supp(\hat{\beta}^{\mathrm{LOG}}) \subseteq supp(\beta^*)$

*(b) For $1 \leq d \leq d+h \leq K^*$ and $\hat{\beta}^{\mathrm{LOG}}_{d+h} \neq 0$,*

$$\delta\frac{|y_{d+h}|}{|y_d|} \leq \frac{\left|\hat{\beta}^{\mathrm{LOG}}_{d+h}\right|}{\left|\hat{\beta}^{\mathrm{LOG}}_{d}\right|} \leq \frac{|y_{d+h}|}{|y_d|}. \tag{2.14}$$

*Proof.* See Appendix A.2.2. □

Equation (2.14) illustrates that the difference in the amount of shrinkage applied by LOG to two elements of different depths does not increase exponentially with the distance $h$ between the two elements. Moreover, the discrepancy in the amount of shrinkage is lower-bounded by a fixed quantity (that, importantly, does not depend on $h$) with high probability. For a fixed $\delta$, the range of $\lambda$ for which this holds is non-empty as $\sigma\sqrt{\log D} \to 0$. Proposition 2 thus establishes that LOG applies a comparable amount of shrinkage at all depths of $\mathcal{D}$. This is corroborated empirically in the middle panels of Figure 2.5 and Figure 2.6.

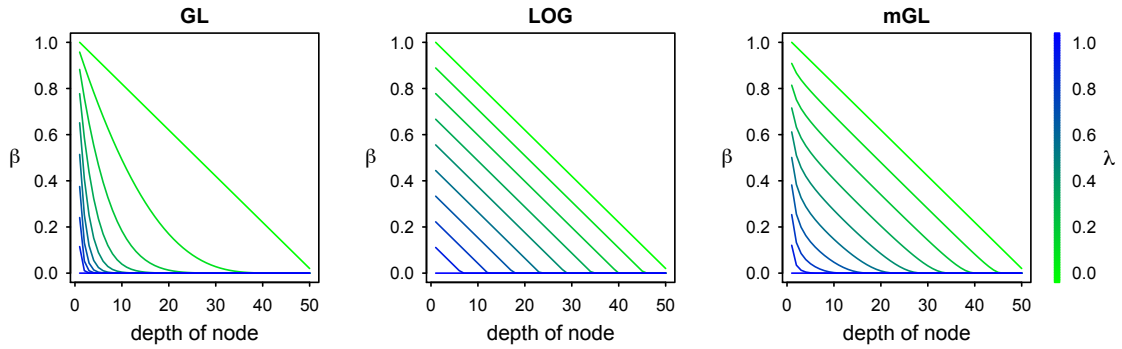


Figure 2.5: The effect of the proximal operator of three regularizers on $\beta_i^* = 1 - \frac{i-1}{D}$: (Left) $\hat{\beta}^{\text{GL}}$, (Middle) $\hat{\beta}^{\text{LOG}}$ and (Right) $\hat{\beta}^{\text{mGL}}$.

To demonstrate how pronounced the differential shrinkage phenomenon of GL is when the DAG depth is large, we plot the elements of $\hat{\beta}^{\text{GL}}$ and $\hat{\beta}^{\text{LOG}}$ when the depth is 50 (Figure 2.3 with $D = 50$). In order to better observe the effect of the proximal operator and thereby better understand the regularizer's influence, we consider a noiseless simuation, i.e., $\sigma = 0$, and therefore $y = \beta^*$. We begin with a situation in which the input to the prox function decays linearly with depth, which might suggest to a statistician good reason to use a regularizer that shrinks elements deep in $\mathcal{D}$ to zero before others:

$$\beta_i^* = 1 - \frac{i-1}{D}, \quad \text{for } i = 1, \ldots, D.$$

The left and middle panels of Figure 2.5 show the proximal operators' outputs for ten equally spaced values of $\lambda$ between 0 and 1. When $\lambda$ is 0 (shown in green), both $\hat{\beta}^{\text{GL}}$ (in the left panel) and $\hat{\beta}^{\text{LOG}}$ (in the middle panel) simply return $y$. As we increase $\lambda$ (shown with increasing levels of blue), one notices a striking difference between the two regularizers. The LOG regularizer preserves the linear nature of the input while the GL regularizer shrinks elements deep in $\mathcal{D}$ to zero at a faster rate than those higher in $\mathcal{D}$. The result is that GL exaggerates the original downward trend in the input.

To balance the aggressive shrinkage of parameters appearing in many groups in the overlapping case, Jenatton et al. (2011a) suggest weighting each parameter in a group differently based on the degree of overlaps existing on the parameter, instead of assigning a single weight to the whole group. In the context of banded covariance estimation, Bien et al. (2016) also find that a better rate of convergence can be obtained using a more elaborate weighting scheme. For a fixed group $g_\ell \in d(\mathcal{D})$, the idea is to apply smaller weights to elements deeper in $\mathcal{D}$. In the directed path graph example, the weight applied to $s_m$ in group $g_\ell = \cup_{m=\ell}^{D} s_m$ is

$$w_{\ell,m} = \frac{1}{m - \ell + 1}, \quad \text{for } 1 \leq \ell \leq m \leq D, \tag{2.15}$$

whereas a more general definition of the weights can be found in Appendix A.5.2. The modified GL (mGL) penalty and the corresponding proximal operator under the general weighting scheme can be denoted as

$$\Omega_{\text{mGL}}^{d(\mathcal{D})}(\beta; \{w_{\ell,m}\}) = \sum_{\ell=1}^{D} \sqrt{\sum_{m=\ell}^{D} w_{\ell,m}^2 \beta_m^2}, \tag{2.16}$$

$$\hat{\beta}^{\text{mGL}} = \underset{\beta \in \mathbb{R}^p}{\arg\min} \left\{ \frac{1}{2} ||y - \beta||_2^2 + \lambda \Omega_{\text{mGL}}^{d(\mathcal{D})}(\beta; \{w_{\ell,m}\}) \right\}. \tag{2.17}$$

In the right panel of Figure 2.5, we see that $\hat{\beta}^{\text{mGL}}$ behaves less aggressively in shrinking elements deep in $\mathcal{D}$. In fact, it appears that GL with general weights mimics the LOG penalty.

Our second example considers a situation in which the raw input is a step function. We take $\beta_i^* = 1_{\{i \leq D/2\}} + 0.5 * 1_{\{i > D/2\}}$ for $i = 1, \ldots, D$. Figure 2.6 shows the effects of the three penalties. We find again that GL creates a strong downward trend whereas LOG preserves the relative sizes of the elements. Again, mGL behaves as a compromise between these two.

In summary, we observe that GL shrinks elements deep in $\mathcal{D}$ more than those

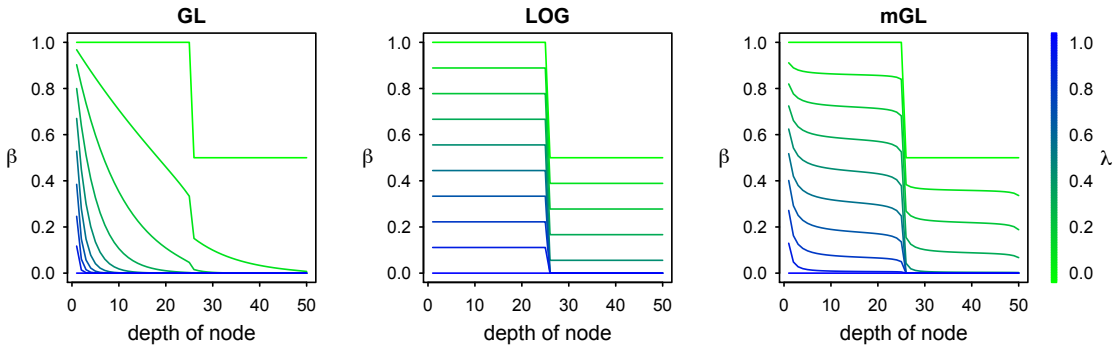


Figure 2.6: The effect of the proximal operator of three regularizers on $\beta_i^* = 1_{\{i \le D/2\}} + 0.5 * 1_{\{i > D/2\}}$: (Left) $\hat{\beta}^{\text{GL}}$, (Middle) $\hat{\beta}^{\text{LOG}}$ and (Right) $\hat{\beta}^{\text{mGL}}$.

high in $\mathcal{D}$. LOG by contrast is able to enforce the HSM constraints without applying differential shrinkage across $\mathcal{D}$. The mGL weighting scheme can effectively balance the aggressiveness of GL and seems reasonable to be used when more aggressive shrinkage is not desired. From a computational standpoint, which is the focus of the next section, this more elaborate weight structure complicates the computation of the proximal operator. Meanwhile, in some cases when the true model is sufficiently sparse, the GL approach, which favors simpler models, may serve a better role. Users should be aware of the difference among these frameworks and consequences, and choose a suitable approach based on their applications.

## 2.4 Computation

Given that both $\Omega_{\text{GL}}^{d(\mathcal{D})}$ and $\Omega_{\text{LOG}}^{a(\mathcal{D})}$ can be used in HSM, we would like to compare them from a computational perspective. Problems (2.6) and (2.8) are nonsmooth convex optimization problems, and proximal gradient methods (Nesterov, 2013; Beck and Teboulle, 2009) are well-suited to such problems, especially when the

non-differentiable part's proximal operator can be efficiently evaluated. We suppose that $F$ is differentiable and that $\nabla F$ is Lipschitz-continuous with constant $L$. In its simplest form, the proximal gradient method iteratively computes (for $k = 0, 1, 2, \ldots$)

$$\beta^{k+1} \leftarrow \arg\min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{2} \left\| \beta - \left( \beta^k - \frac{1}{L} \nabla F(\beta^k) \right) \right\|_2^2 + \lambda \Omega(\beta) \right\},$$

where $\Omega$ can be $\Omega_{\text{GL}}^{d(\mathcal{D})}$ or $\Omega_{\text{LOG}}^{a(\mathcal{D})}$. In words, at each step of the algorithm, the standard gradient descent step for minimizing $F$ is modified by applying the penalty $\lambda\Omega$'s proximal operator. It follows that an important computational benchmark lies in how efficiently the proximal operators, defined in (2.9) and (2.10), can be solved.

The proximal operator of GL when there are overlapping groups is usually solved via the dual problem (Boyd and Vandenberghe, 2004). As described in Jenatton et al. (2011b), a dual of the proximal operator of (2.1) is given by

$$\min_{\{\eta^{(g)} \in \mathbb{R}^p\}_{g \in \mathcal{G}}} \left\{ \frac{1}{2} \left\| y - \sum_{g \in \mathcal{G}} \eta^{(g)} \right\|_2^2 \quad \text{s.t. } \|\eta^{(g)}\|_2 \leq \lambda w_g \text{ and } \eta_{g^c}^{(g)} = 0 \text{ for } g \in \mathcal{G} \right\}.$$

Given a solution $\{\hat{\eta}^{(g)}\}_{g \in \mathcal{G}}$, it can be shown that $\text{Prox}_{\text{GL}}^{\mathcal{G}}(y; \lambda, w) = y - \sum_{g \in \mathcal{G}} \hat{\eta}^{(g)}$. The separable structure of the constraints suggests using block coordinate descent (BCD, Tseng 2001) to solve for $\{\hat{\eta}^{(g)}\}_{g \in \mathcal{G}}$. Algorithm 1 has the details of implementation.

In the special case that $\mathcal{G} = d(\mathcal{D})$ and $\mathcal{D}$ is a tree, Jenatton et al. (2011b) proves the remarkable result that the `while` loop in Algorithm 1 will terminate in one pass, as long as the pass of BCD over $g \in d(\mathcal{D})$ proceeds from innermost groups outward (i.e., from children to parents). The implication of this result is that when $\mathcal{D}$ is a tree, the proximal operator is essentially available in a closed form. Its computational complexity in this situation is $O(p)$, where $p$ is the dimension

**Algorithm 1** BCD in the Dual for Solving the Proximal Operator of $\Omega_{\text{GL}}^{\mathcal{G}}$

**Input:** $y, w, \lambda, \mathcal{G}$.
**Require:** $\lambda \geq 0, w_g > 0 \; \forall g \in \mathcal{G}$.
1: $\eta^{(g)} = 0 \in \mathbb{R}^p$ for all $g \in \mathcal{G}$
2: $\beta = y$
3: **while** stopping criterion not reached **do**
4: **for** $g \in \mathcal{G}$ **do**
5: $\beta \leftarrow \beta + \eta^{(g)}$
6: $\eta^{(g)} \leftarrow \frac{\lambda w_g \beta_g}{\|\beta_g\|_2}$
7: $\beta \leftarrow \beta - \eta^{(g)}$
8: **end for**
9: **end while**
**Output:** $\beta$

of $\beta$. By contrast, there is no known algorithm that solves the proximal operator of $\Omega_{\text{LOG}}^{a(\mathcal{D})}$ in a closed form under a tree structure. Several iterative methods have been used to solve (2.10), including cyclic projection (Villa et al., 2014) and BCD (Obozinski et al., 2011). In Section 2.4.1, we review a commonly-used BCD approach for solving (2.10). In Section 2.4.2, we derive a new closed-form algorithm for solving (2.10) when $\mathcal{D}$ is a directed path graph. Finally, in Section 2.4.3, we leverage this new result to develop a more efficient algorithm for evaluating $\text{Prox}_{\text{LOG}}^{a(\mathcal{D})}$ for general DAGs $\mathcal{D}$.

### 2.4.1 Naive BCD for LOG

By definition of the LOG penalty (2.2), its proximal problem can be rewritten in terms of the latent variables:

$$\min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{2} \|y - \beta\|_2^2 + \lambda \Omega_{\text{LOG}}^{\mathcal{G}}(\beta; w) \right\}$$

$$\Leftrightarrow \quad \min_{\{v^{(g)} \in \mathbb{R}^p\}_{g \in \mathcal{G}}} \left\{ \frac{1}{2} \left\| y - \sum_{g \in \mathcal{G}} v^{(g)} \right\|_2^2 + \lambda \sum_{g \in \mathcal{G}} w_g \left\| v^{(g)} \right\|_2 \quad \text{s.t.} \quad v_{g^c}^{(g)} = 0 \right\}.$$

In this parametrization, the penalty term naturally separates into blocks defined by the latent variables, and one can use BCD, cycling over the latent variable vectors (Obozinski et al., 2011). Algorithm 2 provides the details of this approach, which we refer to as *naive BCD*.

**Algorithm 2** Naive BCD for Solving the Proximal Operator of $\Omega^{\mathcal{G}}_{\text{LOG}}$

**Input:** $y, w, \lambda, \mathcal{G}$.
**Require:** $\lambda \geq 0, w_g > 0 \; \forall g \in \mathcal{G}$.
$v^{(g)} = 0 \in \mathbb{R}^p$ for all $g \in \mathcal{G}$
$\beta = 0 \in \mathbb{R}^p$
**while** stopping criterion not reached **do**
    **for** $g \in \mathcal{G}$ **do**
        $\beta \leftarrow \beta - v^{(g)}$
        $v^{(g)} \leftarrow S_G\left(y_g - \beta_g, \lambda w_g\right)$
        $\beta \leftarrow \beta + v^{(g)}$
    **end for**
**end while**
**Output:** $\beta$

The complexity per cycle of both Algorithm 1 and Algorithm 2 is $O\left(\sum_{g \in \mathcal{G}} |g|\right)$. Recalling that in HSM, for LOG, $\mathcal{G} = a(\mathcal{D})$ contains all ancestor sets whereas for GL, $\mathcal{G} = d(\mathcal{D})$ contains all descendant sets. It is straightforward to observe that $a(\mathcal{D})$ and $d(\mathcal{D})$ have equal numbers of nodes in total. Assuming $|s_i|$ has the same magnitude across $i = 1, \ldots, N$, we see Algorithm 1 and Algorithm 2 require the same order of computation per cycle for general DAGs $\mathcal{D}$.

In the next section, we focus on the case in which $\mathcal{D}$ is a directed path graph and present a new algorithm that exactly solves the proximal operator in a finite number of steps. This will allow us to develop a more efficient alternative to naive BCD for general DAGs.

### 2.4.2 Solution of the LOG Prox for a Directed Path Graph

Suppose that $\mathcal{D}$ is a directed path graph with $D$ nodes as shown in Figure 2.3. We present here what can be seen as LOG counterpart to the result of Jenatton et al. (2011b) for GL when $\mathcal{D}$ is a tree. For notational simplicity, we let $s_{i:j}$ denote $\cup_{k=i}^{j} s_k$. Using this notation, the group structure for the LOG penalty $a(\mathcal{D}) = \{s_{1:\ell} : \ell = 1, \ldots, D\}$ (since $s_{1:\ell}$ is the union of all indices contained in $s_i$ that are ancestors of $s_\ell$). A key quantity in Algorithm 3 is

$$f(j,k) = \frac{\|y_{s_{(k+1):j}}\|_2}{\sqrt{w_j^2 - w_k^2}}, \quad \text{for } 0 \leq k < j \leq D.$$

A standard choice for $w_j$ is $|s_{1:j}|^{1/2}$ in which case the denominator becomes $|s_{(k+1):j}|^{1/2}$ and $f(j,k)^2$ can be thought of as the average of $y_\ell^2$ for $\ell \in s_{(k+1):j}$. The algorithm identifies a sequence of knots $0 = k_0 < k_1 < \cdots < k_m \leq D$ with the properties that $k_i$ maximizes $f(\cdot, k_{i-1})$ and that $f(k_i, k_{i-1}) > \lambda$ for $i = 1, \ldots, m$. The knots are the values that $k$ has taken in the algorithm. Interestingly, once the set of knots has been determined, the algorithm is identical to that of the proximal operator of the non-overlapping group lasso with group structure $\{s_{(k_{i-1}+1):k_i}\}_{i=1,\ldots,m} \cup \{s_{1:D} \setminus s_{1:k_m}\}$ and weights $\{\sqrt{w_{k_i}^2 - w_{k_{i-1}}^2}\}_{i=1,\ldots,m} \cup \{\infty\}$. That is, each vector of elements between consecutive knots is separately groupwise soft-thresholded. The choice of knots implies that only the elements in $s_{1:D} \setminus s_{1:k_m}$ are set to zero. We see that the value of $\lambda$ determines the number of knots $m$, but not their location; thus, when solving the proximal operator for a sequence of $\lambda$ values, we only need to compute the knots once.

**Lemma 3.** *Algorithm 3 computes the proximal operator in* (2.10) *for a directed path graph $\mathcal{D}$ of depth $D$ with complexity $O(p + Dm)$, where $m$ is the number of knots determined by the algorithm (not counting the initialization of $k = 0$). In the worst case*

**Algorithm 3** Solve the Proximal Operator of $\Omega_{\text{LOG}}^{a(\mathcal{D})}$ for a Directed Path Graph $\mathcal{D}$

**Input:** $\lambda \geq 0$, $w = (w_1, \cdots, w_D) \in \mathbb{R}^{+D}$, $y \in \mathbb{R}^p$ and $a(\mathcal{D})$.
**Require:** $w_1 < \cdots < w_D$. $\mathcal{D}$ a path of depth $D$.
1: $\beta \leftarrow 0 \in \mathbb{R}^p$
2: $k \leftarrow 0 \in \mathbb{R}$ ▷ "knots" are values $k$ has taken in the algorithm
3: $w_0 \leftarrow 0 \in \mathbb{R}$
4: **while** $k < D$ **do**
5: $K \leftarrow \arg\max_{j:\, j>k} f(j,k)$ ▷ $f(j,k) = \frac{\|y_{s_{(k+1):j}}\|_2}{\sqrt{w_j^2 - w_k^2}}$ for $0 \leq k < j \leq D$
6: **if** $f(K,k) \leq \lambda$ **then**
7: **break**
8: **end if**
9: $\beta_{s_{(k+1):K}} \leftarrow S_G\left(y_{s_{(k+1):K}}, \lambda\sqrt{w_K^2 - w_k^2}\right)$
10: $k \leftarrow K$
11: **end while**
**Output:** $\beta$

*when there are $D$ knots (i.e., $k$ increases by one and the condition in Line 6 is never satisfied), the complexity is $O(p + D^2)$.*

*Proof.* Appendix A.3 proves that the algorithm computes the proximal operator, and Appendix A.4 proves that when the solution has $m$ knots, Algorithm 3 requires $O(p+Dm)$ operations. To attain this complexity, one does not compute the $f(j,k)$ directly as defined in line 5 of the algorithm but rather performs constant time updates to reduce overall computation. □

In Appendix A.5, we show that the computational complexity of computing $\text{Prox}_{\text{GL}}^{d(\mathcal{D})}$ for this same DAG is $O(p + D)$. This means that when $D$ is larger than $p^{1/2}$, computing GL's prox may be more efficient than computing LOG's prox. By contrast, the computational complexity of computing the proximal operator of the modified GL penalty is $O(p + D^2 \log(n))$, given $n$-digit precision is required in using Newton's method for root-finding.

### 2.4.3 Path-Based BCD and ADMM for LOG

In the previous section, we showed that when $\mathcal{D}$ is a directed path graph, (2.10) can be solved extremely efficiently. For a general DAG $\mathcal{D}$, we can exploit this result by partitioning $\mathcal{D}$ into paths and cycling over the paths until convergence. The left panel of Figure 2.7 shows an example in which we partition a DAG into three paths. Let $\mathcal{P}_1, \cdots, \mathcal{P}_L$ be our path decomposition of $\mathcal{D}$. We require that every node in $\mathcal{D}$ belongs to a unique path $\mathcal{P}_\ell$ and that the edges in path $\mathcal{P}_\ell$ all be in $\mathcal{D}$. The path decomposition of $\mathcal{D}$ induces a partition of $a(\mathcal{D})$ into $\mathcal{G}_1, \cdots, \mathcal{G}_L$, where

$$\mathcal{G}_\ell = \{\text{ancestors}(\mathcal{D}; s_i) : s_i \in \mathcal{P}_\ell\}, \quad \text{for } \ell = 1, \ldots, L.$$

The following lemma shows that the LOG penalty for a general DAG can be decomposed into a sum of LOG penalties, each having the simple path structure. This observation can be exploited to suggest an efficient alternative to naive BCD such that the "blocks" in the new approach are defined by the paths.

**Lemma 4.** *Let $\{\mathcal{G}_\ell\}_{\ell=1}^L$ be the partition of $a(\mathcal{D})$ induced by the path decomposition $\mathcal{P}_1, \cdots, \mathcal{P}_L$ of $\mathcal{D}$. For a convex smooth loss function $F(\beta)$, Problem* (2.8) *can be equivalently solved with*

$$\min_{\{\beta^{(\ell)} \in \mathbb{R}^p\}_{\ell=1}^L} \left\{ F\left(\sum_{\ell=1}^L \beta^{(\ell)}\right) + \lambda \sum_{\ell=1}^L \Omega_{\text{LOG}}^{\mathcal{G}_\ell}(\beta^{(\ell)}; w_{\mathcal{P}_\ell}) \quad \text{s.t.} \quad \text{supp}(\beta^{(\ell)}) \subseteq \bigcup_{g \in \mathcal{G}_\ell} g \right\}, \tag{2.18}$$

*where $w_{\mathcal{P}_\ell} = \{w_g : g \in \mathcal{G}_\ell\}$ for $\ell = 1, \ldots, L$.*

*Proof.* See Appendix A.6. □

Problem (2.18) satisfies the necessary conditions for BCD on $\beta^{(\ell)}$ to converge (Tseng, 2001). For solving the proximal problem (2.10) where $F(\beta) = \frac{1}{2}||y-\beta||_2^2$, Al-

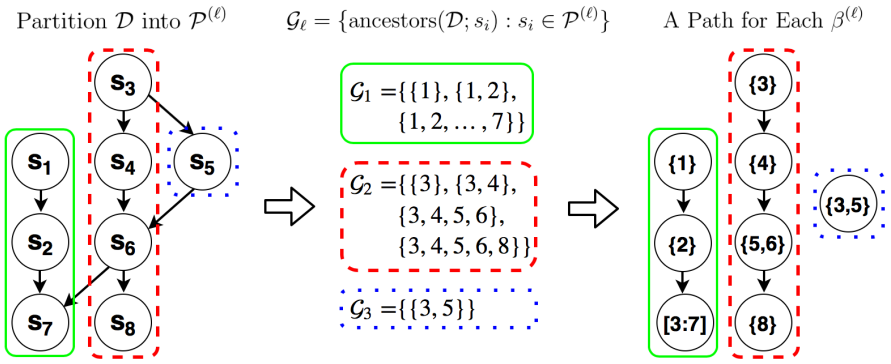


Figure 2.7: Let $s_i = \{i\}$ for $i \in \{1, \cdots, 8\}$. (Left) $a(\mathcal{D})$ is decomposed into 3 path graphs: $\mathcal{P}^{(1)}$ (in green solid contour), $\mathcal{P}^{(2)}$ (in red dashed contour) and $\mathcal{P}^{(3)}$ (in blue dotted contour). (Middle) The partition of $\mathcal{G} = a(\mathcal{D})$: $\mathcal{G}_1$, $\mathcal{G}_2$ and $\mathcal{G}_3$ (colored accordingly). (Right) $a(\mathcal{D})$ can be thought of as three separate path graphs on a new set of nodes, with parameter assignments shown inside each node: (in green solid contour) $supp(\beta^{(1)}) \subseteq \{1, \cdots, 7\}$, (in red dashed contour) $supp(\beta^{(2)}) \subseteq \{3, 4, 5, 6, 8\}$ and (in blue dotted contour) $supp(\beta^{(3)}) \subseteq \{3, 5\}$.

gorithm 4 presents what we call *path-based BCD*. The value of this reparametrization is that each block update can be efficiently solved using Algorithm 3. When there are long paths in $\mathcal{D}$, the path-based BCD can make much faster progress compared to naive BCD since we are able to jointly minimize over all nodes in the path rather than settle for slow incremental progress. The decomposition of a DAG into paths is non-unique and the choice of path decomposition will affect efficiency. Algorithm 7 in Appendix A.7 presents a simple greedy approach that attempts to break $\mathcal{D}$ into long paths. The *path-based BCD* is implemented in the R package `hsm` that is available on CRAN.

Clearly, the greatest efficiency gains for path-based BCD are to be expected when $\mathcal{D}$ can be decomposed into a small number of long path graphs. By contrast, the least favorable case for the path-based BCD is when $\mathcal{D}$ is a depth-two tree since this structure does not have any long paths. The upper panel of Figure 2.8 shows these two trees along with a binary tree, which represents a choice for $\mathcal{D}$ between these two extremes. We perform simulations for these three choices

**Algorithm 4** Path-based BCD for Solving the Proximal Operator of $\Omega_{\text{LOG}}^{a(\mathcal{D})}$

**Input:** $y \in \mathbb{R}^p, w, \lambda, \mathcal{D}$, and a path-decomposition $\{\mathcal{P}_\ell\}_{\ell=1}^L$ of $\mathcal{D}$.

Generate $\mathcal{G}_\ell$ from $a(\mathcal{D})$ and $\{\mathcal{P}_\ell\}$.
$S_\ell \leftarrow \cup_{g \in \mathcal{G}_\ell} g$ for $\ell = 1, \ldots, L$
$\beta^{(\ell)} \leftarrow 0 \in \mathbb{R}^p$ for $\ell = 1, \ldots, L$
$\beta \leftarrow 0 \in \mathbb{R}^p$
**while** stopping criterion not reached **do**
    **for** $\ell \in [1 : L]$ **do**
        $\beta \leftarrow \beta - \beta^{(\ell)}$
        $\beta_{S_\ell}^{(\ell)} \leftarrow \text{Prox}_{\text{LOG}}^{\mathcal{G}_\ell}(y_{S_\ell} - \beta_{S_\ell}; \lambda, w_{\mathcal{P}_\ell})$ ▷ solved using Algorithm 3
        $\beta \leftarrow \beta + \beta^{(\ell)}$
    **end for**
**end while**

**Output:** $\beta$

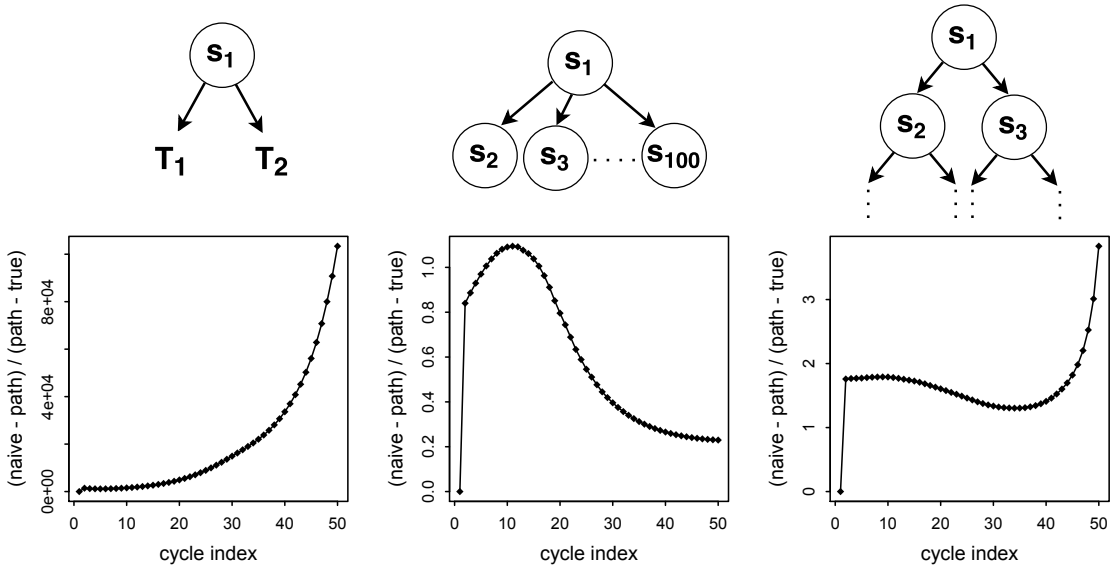


Figure 2.8: (Top) Tree structures for example 1, 2 and 3, respectively. On top left, $T_1$ and $T_2$ are path graphs of length 50 and 49, respectively. (Bottom) Plot of ratio of the difference in objective values of the two BCDs and the difference in objective value of the path-based BCD and the "truth", evaluated at each cycle and averaged over 20 realizations, with the corresponding tree above it.

of $\mathcal{D}$ to compare the rate of change of objective values using both BCD schemes. In the first example (upper left panel of Figure 2.8), $T_1$ and $T_2$ are path graphs of length 50 and 49, respectively, and each node has $|s_i| = 5$ (for a total of $p = 500$ parameters); in the second example (upper middle panel), we again have $|s_i| = 5$ (and $p = 500$); in the third example (upper right panel), we take a binary tree of depth 9, with $|s_i| = 1$ ($p = 2^9 - 1 = 511$). In all cases, we take $\lambda = 0.1$ and $w_g = |g|^{1/2}$.

For each $\mathcal{D}$, we randomly draw 20 samples of $y$ from $N_p(\mu = 0, \Sigma = 4I_p)$, and use both methods to solve (2.10) at each $y$. The bottom panels of Figure 2.8 show the evolution over 50 cycles the ratio of the difference in objective values of the two BCDs and the difference in objective value of the path-based BCD and the "truth", evaluated at each cycle and averaged over 20 realizations. For each $(\mathcal{D}, y)$ pair, the objective evaluated at true parameter value is estimated with the minimum objective value computed over all the cycles of the two methods. All three curves are above zero after the starting point, indicating the naive approach is slower. In the most favorable case for path-based BCD (example 1), we see great advantage of using path-based BCD since the curve is in a much higher magnitude than the other two. As expected, path-based BCD has minor advantage over naive BCD in the depth-two tree case. For example, in the second cycle of the middle panel, the ratio takes value 0.8, meaning that (naive objective - true objective) is 80% larger than (path objective - true objective). In a non-extreme case represented by binary tree, path-based BCD still converges faster than naive BCD. For a more general $F(\beta) = \frac{1}{2}\|y - \mathbf{X}\beta\|_2^2$ in (2.8), Lemma 4 can be used to suggest an efficient alternating direction method of multipliers (ADMM, Boyd et al. 2011) approach:

**Lemma 5** (Path-based ADMM)**.** *Let* $\{\mathcal{G}_\ell\}_{\ell=1}^L$ *be the partition of* $a(\mathcal{D})$ *induced by the*

*path decomposition* $\mathcal{P}_1, \cdots, \mathcal{P}_L$ *of* $\mathcal{D}$*. For* $y \in \mathbb{R}^n$ *and* $\mathbf{X} \in \mathbb{R}^{n \times p}$*, Problem* (2.8) *with* $F(\beta) = \frac{1}{2}\|y - \mathbf{X}\beta\|_2^2$ *can be equivalently solved using ADMM on the following problem:*

$$\min_{\{\beta^{(\ell)} \in \mathbb{R}^p, \gamma^{(\ell)} \in \mathbb{R}^p\}_{\ell=1}^L} \frac{1}{2}\left\| y - \mathbf{X}\sum_{\ell=1}^L \gamma^{(\ell)} \right\|_2^2 + \lambda \sum_{\ell=1}^L \Omega_{\mathrm{LOG}}^{\mathcal{G}_\ell}(\beta^{(\ell)}; w_{\mathcal{P}_\ell})$$
$$\text{s.t.} \quad \beta^{(\ell)} = \gamma^{(\ell)} \text{ and } \mathrm{supp}(\beta^{(\ell)}) \subseteq \bigcup_{g \in \mathcal{G}_\ell} g =: g^{(\ell)} \quad \forall \ell = 1, \ldots, L. \tag{2.19}$$

*The ADMM iterates among the following three steps and uses Algorithm 4 to solve Step (2).*

*(1)*
$$\hat{\gamma}_{|g^{(\ell)}}^{(\ell)} \leftarrow \hat{\beta}_{|g^{(\ell)}}^{(\ell)} + \frac{1}{\rho}\hat{u}_{|g^{(\ell)}}^{(\ell)} + \frac{1}{\rho}\mathbf{X}_{|g^{(\ell)}}^T(y - \Delta) \qquad \forall \ell = 1, \ldots, L,$$
$$\text{where } \Delta = \left(I + \frac{1}{\rho}\sum_{\ell} \mathbf{X}_{|g^{(\ell)}}\mathbf{X}_{|g^{(\ell)}}^T\right)^{-1} \sum_{\ell}\left(\mathbf{X}_{|g^{(\ell)}}\left(\hat{\beta}_{|g^{(\ell)}}^{(\ell)} + \frac{1}{\rho}\hat{u}_{|g^{(\ell)}}^{(\ell)}\right) + \frac{1}{\rho}\mathbf{X}_{|g^{(\ell)}}\mathbf{X}_{|g^{(\ell)}}^T y\right).$$

*(2)*
$$\hat{\beta}_{|g^{(\ell)}}^{(\ell)} \leftarrow \mathrm{Prox}_{\mathrm{LOG}}^{\mathcal{G}_\ell}\left(\left(\hat{\gamma}_{|g^{(\ell)}}^{(\ell)} - \frac{1}{\rho}\hat{u}_{|g^{(\ell)}}^{(\ell)}\right); \frac{\lambda}{\rho}, w_{\mathcal{P}_\ell}\right) \qquad \forall \ell = 1, \ldots, L.$$

*(3)*
$$\hat{u}^{(\ell)} \leftarrow \hat{u}^{(\ell)} + \rho\left(\hat{\gamma}^{(\ell)} - \hat{\beta}^{(\ell)}\right) \qquad \forall \ell = 1, \ldots, L.$$

*Proof.* See Appendix A.8. □

## 2.5 Estimating Banded Covariance with LOG

In Section 2.3, we observed that LOG avoids applying differential shrinkage on $\mathcal{D}$ as is in GL. In Section 2.4, we showed that when $\mathcal{D}$ is a directed path graph, the proximal operator can be evaluated in a closed form. In this section, we synthesize these observations in an application to covariance estimation. This example will demonstrate how choosing the LOG penalty leads to an estimator

that achieves the statistical advantages of an existing estimator that requires the more complicated modified GL approach.

Suppose we observe a sample $X^{(1)}, X^{(2)}, \cdots, X^{(n)} \in \mathbb{R}^p$ of independent zero-mean random vectors with true population covariance matrix $\mathbf{\Sigma}^*$. If the $p$ variables have a known ordering, a common assumption is that $\mathbf{\Sigma}^*$ is $K$-banded, meaning that

$$\mathbf{\Sigma}^*_{ij} = 0 \text{ for } |i-j| > K.$$

The sample covariance matrix, $\mathbf{S} = \frac{1}{n}\sum_{i=1}^n (X^{(i)} - \bar{X})(X^{(i)} - \bar{X})^T$ (where $\bar{X} = \frac{1}{n}\sum_{i=1}^n X^{(i)}$), degrades as an estimator of $\mathbf{\Sigma}^*$ as $p$ increases; when $\mathbf{\Sigma}^*$ is (or could be reasonably approximated as) a banded matrix, banded estimators are preferable. It is straightforward to see that banded estimation of a matrix is an instance of HSM: Take $\mathcal{D}$ to be a directed path graph, such as that depicted in Figure 2.3, where

$$s_m = \{ij \in \{1, \ldots, p\}^2 : |i-j| = m\}, \text{ for } m = 1, \ldots, p-1,$$

is the "subdiagonal" of elements that are $m$ away from the main diagonal. Bandedness of $\mathbf{\Sigma}$ can then be expressed as $\mathbf{\Sigma}_{s_\ell} = 0 \implies \mathbf{\Sigma}_{s_m} = 0$ for any $m > \ell$.

Bien et al. (2016) propose "convex banding" estimators, which, in the terminology of our chapter, correspond to

$$\hat{\mathbf{\Sigma}}^{\text{GL}} = \underset{\mathbf{\Sigma} \in \mathbb{R}^{p \times p}}{\arg\min} \left\{ \frac{1}{2} \|\mathbf{S} - \mathbf{\Sigma}\|_F^2 + \lambda \Omega_{\text{GL}}^{d(\mathcal{D})}(\mathbf{\Sigma}^-; w) \right\}, \quad \text{with } w_\ell = \sqrt{|s_\ell|}$$

being the weight on the group $s_{\ell:D}$, and

$$\hat{\mathbf{\Sigma}}^{\text{mGL}} = \underset{\mathbf{\Sigma} \in \mathbb{R}^{p \times p}}{\arg\min} \left\{ \frac{1}{2} \|\mathbf{S} - \mathbf{\Sigma}\|_F^2 + \lambda \Omega_{\text{mGL}}^{d(\mathcal{D})}(\mathbf{\Sigma}^-; \tilde{w}) \right\}, \quad \text{with } \tilde{w}_{\ell,m} = \sqrt{|s_\ell|}/(m - \ell + 1)$$

being the weight on $s_m$ within the group $s_{\ell:D}$, where $\mathbf{\Sigma}^-$ denotes the matrix $\mathbf{\Sigma}$ but with zeros on its main diagonal. We recognize these as the proximal operators of the two penalties. Bien et al. (2016) prove that both estimators can recover

the true bandwidth with high probability; however, only $\hat{\mathbf{\Sigma}}^{\text{mGL}}$, and not $\hat{\mathbf{\Sigma}}^{\text{GL}}$, is shown to attain (up to a logarithmic factor) the minimax rate of convergence in Frobenius norm over a certain class of covariance matrices. They suggest that it is the overly aggressive shrinkage of subdiagonals far from the main diagonal (i.e., $s_m$ deep in $\mathcal{D}$) that prevents them from getting a similar rate for $\hat{\mathbf{\Sigma}}^{\text{GL}}$.

In light of our observation in Section 2.3 that LOG applies a comparable amount of shrinkage at all depths of $\mathcal{D}$, we investigate in this section whether a banded covariance estimator based instead on LOG can match the performance of $\hat{\mathbf{\Sigma}}^{\text{mGL}}$. We will show that this LOG-based covariance estimator does successfully match the statistical performance of $\hat{\mathbf{\Sigma}}^{\text{mGL}}$, and, notably, does not require any modification of the weights as was the case with the GL-based estimator.

### 2.5.1 Defining the Estimator $\hat{\mathbf{\Sigma}}^{\text{LOG}}$

We define $\hat{\mathbf{\Sigma}}^{\text{LOG}}$ as the solution to the following problem:

$$\hat{\mathbf{\Sigma}}^{\text{LOG}} = \underset{\mathbf{\Sigma} \in \mathbb{R}^{p \times p}}{\arg\min} \left\{ \frac{1}{2} \|\mathbf{\Sigma} - \mathbf{S}\|_F^2 + \lambda \Omega_{\text{LOG}}^{a(\mathcal{D})}(\mathbf{\Sigma}^-; w) \right\}, \quad \text{with } w_m = \sqrt{|s_{1:m}|} \tag{2.20}$$

being the weight on the group $s_{1:m}$. The group structure $a(\mathcal{D})$ is depicted in Figure 2.9. A key property of the "convex banding" estimators (Bien et al., 2016) is that they can be evaluated in a single pass over the elements of $\mathbf{S}$. By our result in Section 2.4.2, this advantageous computational property is shared by $\hat{\mathbf{\Sigma}}^{\text{LOG}}$. For completeness, Algorithm 3 in the context of covariance estimation is provided in Algorithm 8 of Appendix A.12.

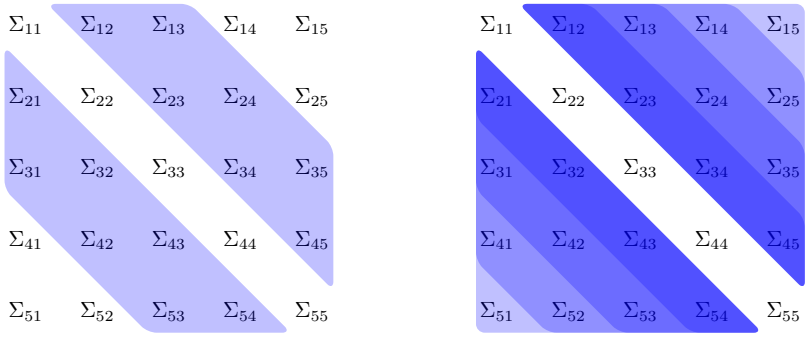


Figure 2.9: (Left) The group $s_{1:2}$; (Right) The nested groups of the form $s_{1:k}$ in $a(\mathcal{D})$.

### 2.5.2 Statistical Properties of $\hat{\mathbf{\Sigma}}^{\text{LOG}}$

We briefly review the statistical assumptions made in Bien et al. (2016), which we will assume hold here as well.

*Assumption* 1. The random vector $X = (X_1, \cdots, X_p)^T \in \mathbb{R}^p$ (which is mean 0 with covariance matrix $\mathbf{\Sigma}^*$) is marginally sub-Gaussian, i.e.,

$$\mathbb{E} \exp(tX_i / \sqrt{\mathbf{\Sigma}^*_{ii}}) \leq \exp(Ct^2)$$

for all $t \geq 0$ and for some $C > 0$. Further, $\max_i |\mathbf{\Sigma}^*_{ii}| \leq M$ for some constant $M > 0$.

*Assumption* 2. The dimension $p$ and sample size $n$ scale as follows: $\gamma_0 \log n \leq \log p \leq \gamma n$ for some $\gamma_0 > 0, \gamma > 0$.

Under these assumptions, it is proved in Lemma 1 of Bien et al. (2016) that the random set

$$\mathcal{A}_x = \left\{ \max_{1 \leq i,j \leq p} \left| \mathbf{S}_{ij} - \mathbf{\Sigma}^*_{ij} \right| \leq x \sqrt{\log p / n} \right\},$$

has high probability for sufficiently large $x$.

**Exact Bandwidth Recovery**

Suppose the true population covariance matrix $\mathbf{\Sigma}^*$ has bandwidth $K$, that is, we have $\mathbf{\Sigma}^*_{s_K} \neq 0$ and $\mathbf{\Sigma}^*_{s_k} = 0$ for $k > K$. Let $\hat{K}$ denote the bandwidth of $\hat{\mathbf{\Sigma}}^{\text{LOG}}$. We show in Theorem 1 and Theorem 2 that under mild conditions our estimator $\hat{\mathbf{\Sigma}}^{\text{LOG}}$ correctly recovers $K$ with high probability.

**Theorem 1.** *If $\lambda \geq x\sqrt{\log p/n}$, then $\hat{K} \leq K$ with high probability.*

*Proof.* See Appendix A.9. □

From Theorem 1 we see that for large enough $\lambda$, $\hat{\mathbf{\Sigma}}^{\text{LOG}}$ will not overestimate $K$. In order for $\hat{\mathbf{\Sigma}}^{\text{LOG}}$ not to underestimate the true bandwidth, we need the nonzero elements of $\mathbf{\Sigma}^*$ to be sufficiently large. In the next theorem, we quantify the signal size by the root-mean-square of the elements of $\mathbf{\Sigma}^*$ in each group of the form $s_{m:K}$ for $m = 1, \ldots, K$.

**Theorem 2.** *Take $\lambda$ as in Theorem 1. If*

$$\min_{1 \leq m \leq K} \frac{\left\| \mathbf{\Sigma}^*_{s_{m:K}} \right\|_F}{\sqrt{|s_{m:K}|}} > 2\lambda, \tag{2.21}$$

*then $\hat{K} \geq K$ with high probability.*

*Proof.* See Appendix A.10. □

Thus, under the above signal strength condition, our LOG-based estimator correctly recovers the bandwidth with high probability. Furthermore, this condition is implied by the corresponding condition appearing in Theorem 4 of Bien et al. (2016). This establishes that the LOG estimator recovers bandwidth at least as well as the "convex banding" estimators.

**Convergence in Frobenius Norm**

In this section we show that $\hat{\mathbf{\Sigma}}^{\text{LOG}}$ achieves, up to a multiplicative logarithmic factor, the optimal rate of convergence in Frobenius norm over the class of $K$-banded covariance matrices $\mathbf{\Sigma}^*$.

**Theorem 3.** *Suppose $\mathbf{\Sigma}^*$ has bandwidth $K$. If $\lambda = x\sqrt{\log p/n}$, then with high probability*

$$\|\hat{\mathbf{\Sigma}}^{\text{LOG}} - \mathbf{\Sigma}^*\|_F^2 \lesssim \frac{pK \log p}{n}, \tag{2.22}$$

*where $\lesssim$ denotes an inequality holding up to a positive multiplicative constant independent of $n$ or $p$.*

*Proof.* See Appendix A.11. □

This rate matches the statistical rate shown for $\hat{\mathbf{\Sigma}}^{\text{mGL}}$, but is noteworthy in that $\hat{\mathbf{\Sigma}}^{\text{LOG}}$ does not require the sophisticated weight structure of $\hat{\mathbf{\Sigma}}^{\text{mGL}}$.

### 2.5.3 Simulation Study

From Section 2.5.2, we see that the estimators $\hat{\mathbf{\Sigma}}^{\text{LOG}}$ and $\hat{\mathbf{\Sigma}}^{\text{mGL}}$ have comparable theoretical properties; moreover, they both share the beneficial computational property that they can be computed in a single pass over the parameters. The more complicated weighting scheme of $\hat{\mathbf{\Sigma}}^{\text{mGL}}$ requires solving a one-dimensional line search for every subdiagonal whereas all operations in computing $\hat{\mathbf{\Sigma}}^{\text{LOG}}$ are very simple. We now further our comparison in two empirical studies. We consider two patterns for $\mathbf{\Sigma}^*$: a *moving-average pattern* and a *stair*

*pattern*. The moving-average pattern corresponds to a downward linear decay in subdiagonal values:

$$\mathbf{\Sigma}^* = toeplitz\left(\left(1, \frac{K-1}{K}, \ldots, \frac{1}{K}, 0_{p-K}\right)\right) \tag{2.23}$$

where $toeplitz(v)$ denotes a symmetric Toeplitz matrix with $v \in \mathbb{R}^p$ being the first column. The stair pattern, as its name suggests, adds flatness to the decay by introducing a "staircase" pattern in $\mathbf{\Sigma}^*$. We construct $\mathbf{\Delta} \in \mathbb{R}^{p \times p}$ as

$$\mathbf{\Delta} = toeplitz\left(\left(1_{\frac{K}{5}}, 0.8 * 1_{\frac{K}{5}}, 0.6 * 1_{\frac{K}{5}}, 0.4 * 1_{\frac{K}{5}}, 0.2 * 1_{\frac{K}{5}}, 0_{p-K}\right)\right)$$

and define

$$\mathbf{\Sigma}^* = \mathbf{\Delta} + (0.01 - \lambda_{min}(\mathbf{\Delta}))_+ I_p \tag{2.24}$$

so that the minimum eigenvalue of $\mathbf{\Sigma}^*$ is at least 0.01.

For both studies, we simulate 50 samples of size 50 with a given $\mathbf{\Sigma}^*$, where each sample is denoted as $\left\{X^{(i)} \overset{i.i.d.}{\sim} N_p(0, \mathbf{\Sigma}^*) \text{ for } i = 1, \ldots, 50\right\}$. A sample covariance $\mathbf{S}_j$ is computed with the $j$th sample. In terms of evaluating performance, we use *mean-squared error* as the metric of comparison:

$$MSE(\lambda) = \frac{1}{50} \sum_{j=1}^{50} \left\| \hat{\mathbf{\Sigma}}(\lambda, \mathbf{S}_j) - \mathbf{\Sigma}^* \right\|_F^2 / p. \tag{2.25}$$

In the first study, we investigate to what extent the rate of $\hat{\mathbf{\Sigma}}^{\text{LOG}}$ derived in Theorem 3 in terms of $K$ and $p$ holds in practice. We simulate under the model used in Section 5.1.1 of Bien et al. (2016). In particular, we take $\lambda_{theory} = 2\sqrt{\log p / n}$ and simulate with $\mathbf{\Sigma}^*$ in (2.23) for $p \in \{500, 1000, 2000\}$. At each $p$, we vary $K$ over 10 values equally spaced between 10 and 500. In agreement with Theorem 3, the left panel of Figure 2.10 shows (for three values of $p$) an approximate linear dependence of $K$ on squared Frobenius norm. The right panel supports the $p$

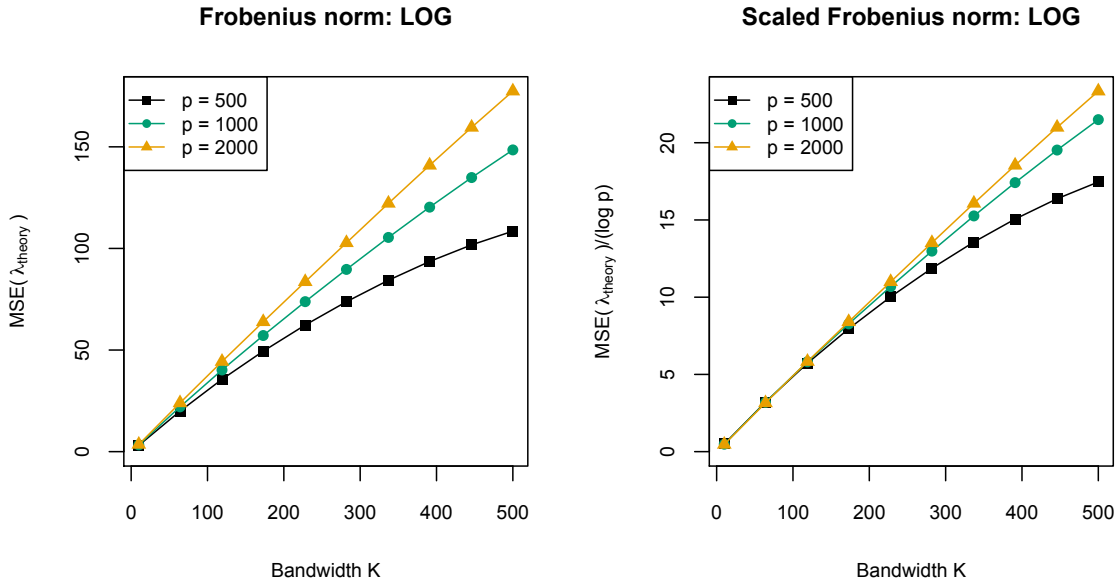


Figure 2.10: (Left) $MSE(\lambda_{theory})$ and (Right) $MSE(\lambda_{theory})/\log p$ as a function of $K$ for $\hat{\mathbf{\Sigma}}^{\text{LOG}}$ where $\lambda_{theory} = 2\sqrt{\log p/n}$ .

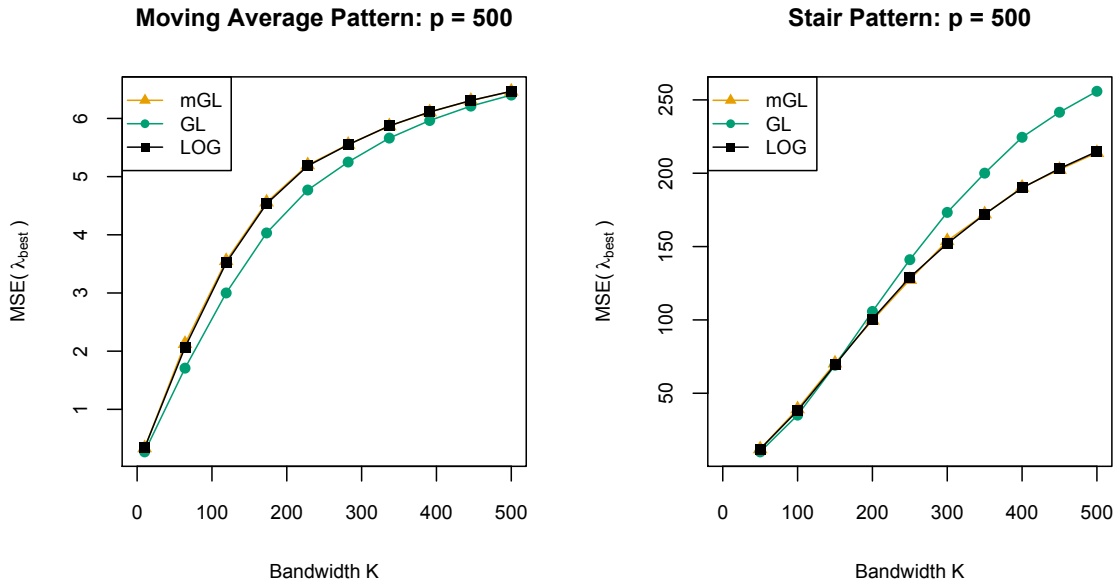


Figure 2.11: For the three estimators $\left(\hat{\mathbf{\Sigma}}^{\text{mGL}}, \hat{\mathbf{\Sigma}}^{\text{GL}}, \hat{\mathbf{\Sigma}}^{\text{LOG}}\right)$, $MSE(\lambda_{best})$ as a function of $K$ under the moving average pattern (Left) and the stair pattern (Right) where $\lambda_{best} = \arg\min_{\lambda \in \Lambda} MSE(\lambda)$.

dependence of Theorem 3 since we find that the three curves line up when we scale the squared Frobenius norm by $p \log p$.

In the second study, we compare the empirical performance of $\hat{\mathbf{\Sigma}}^{\text{GL}}$, $\hat{\mathbf{\Sigma}}^{\text{mGL}}$, and $\hat{\mathbf{\Sigma}}^{\text{LOG}}$ over the two patterns for $\mathbf{\Sigma}^*$ at $p = 500$ and for various $K$. In contrast to the previous study, where we used the theoretically justified $\lambda_{theory}$ of the form $x\sqrt{\log p/n}$, in this study we use

$$\lambda_{best} = \arg\min_{\lambda \in \Lambda} MSE(\lambda) \tag{2.26}$$

where $\Lambda$ is a grid of 50 values equally spaced on the log scale. The quantity $MSE(\lambda_{best})$ is an estimate of $\min_\lambda \mathbb{E}\|\hat{\mathbf{\Sigma}}(\lambda) - \mathbf{\Sigma}^*\|_F^2/p$ and provides a view of the best obtainable performance of each method.

We first consider the moving-average pattern described in (2.23) for $\mathbf{\Sigma}^*$ with $K$ varying over 10 equally-spaced values between 10 and 500. The left panel of Figure 2.11 shows how $MSE(\lambda_{best})$ varies with $K$ for the three methods. We notice that $\hat{\mathbf{\Sigma}}^{\text{GL}}$ outperforms $\hat{\mathbf{\Sigma}}^{\text{mGL}}$ and $\hat{\mathbf{\Sigma}}^{\text{LOG}}$ at all $K$. In addition, $\hat{\mathbf{\Sigma}}^{\text{mGL}}$ and $\hat{\mathbf{\Sigma}}^{\text{LOG}}$ appear to perform similarly. It is striking to compare the scale of the y-axis in the left panel of Figure 2.10 to that of Figure 2.11. Figure 2.10 shows the performance of $\hat{\mathbf{\Sigma}}^{\text{LOG}}$ with $\lambda_{theory} = 2\sqrt{\log p/n}$, which while motivated by theory, is evidently far from the optimal choice of $\lambda$ in terms of $MSE$. The sublinear curve seen in Figure 2.11 is again a reminder that the theory is about $\lambda = x\sqrt{\log p/n}$ and not about $\lambda_{best}$.

The second pattern we consider for $\mathbf{\Sigma}^*$ is the stair pattern described in (2.24) with $K$ varying over 10 equally-spaced values between 50 and 500. As shown in the right panel of Figure 2.11, all three estimators achieve much larger error than in the moving average case. When $K$ is small ($K < 200$), $\hat{\mathbf{\Sigma}}^{\text{GL}}$ beats $\hat{\mathbf{\Sigma}}^{\text{mGL}}$ and $\hat{\mathbf{\Sigma}}^{\text{LOG}}$, but by a small amount. When $K$ becomes larger, both $\hat{\mathbf{\Sigma}}^{\text{mGL}}$ and $\hat{\mathbf{\Sigma}}^{\text{LOG}}$

outperform $\hat{\mathbf{\Sigma}}^{\text{GL}}$. We again see similar performance between $\hat{\mathbf{\Sigma}}^{\text{mGL}}$ and $\hat{\mathbf{\Sigma}}^{\text{LOG}}$. The relative performance of these three methods in these two scenarios suggests that LOG and mGL perform very similarly and that it is difficult to say in general whether these perform better or worse than GL.

Since we are estimating a covariance matrix, we are also interested in getting a positive semidefinite (PSD) estimate. For the stair pattern, we find in simulation that these three estimators are always PSD. By contrast, in the moving-average example, we find that the probability of each estimator being PSD at each method's $\lambda_{best}$ varies with $K$ (see Figure A.1 of Appendix A.13). We find that the probability that $\hat{\mathbf{\Sigma}}^{\text{GL}}$ is PSD decreases to 0 as $K$ increases to $p$. For $\hat{\mathbf{\Sigma}}^{\text{mGL}}$ and $\hat{\mathbf{\Sigma}}^{\text{LOG}}$, the $K$ dependence is less simple; for large $K$, the probability that they are PSD is approximately 80%, but for moderate $K$, we find the probability drops to as low as 20%. If positive definiteness is important in a given application, one could modify (2.20) to include a PSD constraint as is done in Problem (2.3) of Bien et al. (2016).

## 2.6 Conclusion

In this chapter, we focus on hierarchical sparse modeling, a structure that arises in a wide array of statistical problems. In particular, we investigate the differences between two convex penalties, GL and LOG, that have been used in this context for identical purposes but until now have not been systematically compared for HSM. We highlight a phenomenon of GL in which parameters embedded deep within the HSM's DAG are more aggressively regularized than those that are less deeply embedded. We find that this phenomenon may have nega-

tive statistical consequences for GL—both theoretical and empirical—when the DAG has deep nodes and the true model is not very sparse. While a modification of GL is possible to curb this over-aggressiveness of GL (Jenatton et al., 2011a; Bach et al., 2012; Bien et al., 2016), doing so complicates the computation and makes for a more difficult to describe estimator. By contrast, we show that using LOG fulfills our goal without any additional complication and performs, both in practice and in theory, very similarly to the modified GL penalty. In the special case that the DAG is a path, we derive a closed-form expression for the proximal operator of LOG that can be seen as the LOG counterpart to a result of Jenatton et al. (2011b) about the GL penalty. Having this closed form makes computation extremely efficient for directed path graphs, and we leverage this efficiency to general DAGs and more general problems by proposing path-based BCD and path-based ADMM algorithms. We show in simulation that the path-based BCD algorithm converges in fewer passes over the parameters than the standard BCD approach for LOG.

As an application of these ideas to statistics, we show how the recent "convex banding" covariance estimator of Bien et al. (2016) could have instead been formulated with an LOG penalty. We show that our LOG-based estimator attains the same convergence and recovery results as the mGL-based appoach in Bien et al. (2016) and in simulation performs extremely similarly as well. The advantage of our LOG estimator is that it is easier to describe and compute.

CHAPTER 3

# A TREE-BASED RARE FEATURE SELECTION FRAMEWORK IN HIGH DIMENSIONS

*Portions of this chapter were published in Yan and Bien (2018).*

## 3.1 Introduction

The assumption of parameter sparsity plays an important simplifying role in high-dimensional statistics. However, this chapter is focused on sparsity in the data itself, which actually makes estimation more challenging. In many modern prediction problems, the design matrix has many columns that are highly sparse. This arises when the features record the frequency of events (or the number of times certain properties hold). While a small number of these events may be common, there is typically a very large number of rare events, which correspond to features that are zero for nearly all observations. We call these predictors *rare features*. Rare features are in fact extremely common in many modern data sets. For example, consider the task of predicting user behavior based on past website visits: Only a small number of sites are visited by a lot of the users; all other sites are visited by only a small proportion of users. As another example, consider text mining, in which one makes predictions about documents based on the terms used. A typical approach is to create a document-term matrix in which each column encodes a term's frequency across documents. In such domains, it is often the case that the majority of the terms appear very infrequently across the documents; hence the corresponding columns in the document-term matrix are very sparse (e.g., Forman 2003; Huang 2008; Liu et al. 2010; Wang et al. 2010). In Section 3.6, we study a text dataset with more than

200 thousand reviews crawled from `https://www.tripadvisor.com`. Our goal is to use the adjectives in a review to predict a user's numerical rating of a hotel. As shown in the right panel of Figure 3.5, the distribution of adjective density, defined as the proportion of documents containing an adjective, is extremely right-skewed, with many adjectives occurring very infrequently in the corpus. In fact, we find that more than 95% of the 7,787 adjectives appear in less than 5% of the reviews. It is common practice to simply discard rare terms,[1] which may mean removing most of the terms (e.g., Forman 2003; Huang 2008; Liu et al. 2010; Wang et al. 2010).

Rare features also arise in various scientific fields. For example, microbiome data measure the abundances of a large number of microbial species in a given environment. Researchers use next generation sequencing technologies, clustering these reads into "operational taxonomic units" (OTUs), which are roughly thought of as different species of microbe (e.g., Schloss et al. 2009; Caporaso et al. 2010). In practice, many OTUs are rare, and researchers often aggregate the OTUs to genus or higher levels (e.g., Zhang et al. 2012; Chen et al. 2013; Xia et al. 2013; Lin et al. 2014; Randolph et al. 2015; Shi et al. 2016; Cao et al. 2017) or with unsupervised clustering techniques (e.g. McMurdie and Holmes 2013; Wang and Zhao 2017b) to create denser features. However, even after this step, a large portion of these aggregated OTUs are still found to be too sparse and thus are discarded (e.g., Zhang et al. 2012; Chen et al. 2013; Shi et al. 2016; Wang and Zhao 2017b). The rationale for this elimination of rare OTUs is that there needs to be enough variation among samples for an OTU to be successfully estimated in a statistical model (Ridenhour et al., 2017).

[1]For example, in the `R` text mining library `tm` (Feinerer and Hornik, 2017), `removeSparseTerms` is a commonly used function for removing any terms with sparsity level above a certain threshold.

The practice of discarding rare features is wasteful: a rare feature should not be interpreted as an unimportant one since it can be highly predictive of the response. For instance, using the word "ghastly" in a hotel review delivers an obvious negative sentiment, but this adjective appears very infrequently in TripAdvisor reviews. Discarding an informative word like "ghastly" simply because it is rare clearly seems inadvisable. To throw out over half of one's features is to ignore what may be a huge amount of useful information.

Even if rare features are not explicitly discarded, many existing variable selection methods are unable to select them. The challenge is that with limited examples there is very little information to identify a rare feature as important. Theorem 4 shows that even a single rare feature can render ordinary least squares (OLS) inconsistent in the classical limit of infinite sample size and fixed dimension.

To address the challenge posed by rare features, we propose in this work a method for forming new aggregated features which are less sparse than the original ones and may be more relevant to the prediction task. Consider the following features, which represent the frequency of certain adjectives used in hotel reviews:

- $\boldsymbol{X}_{\text{worrying}}, \boldsymbol{X}_{\text{depressing}}, \ldots, \boldsymbol{X}_{\text{troubling}}$,
- $\boldsymbol{X}_{\text{horrid}}, \boldsymbol{X}_{\text{hideous}}, \ldots, \boldsymbol{X}_{\text{awful}}$.

While both sets of adjectives express negative sentiments, the first set (which might be summarized as "worry") seems more mild than the second set (which might be summarized as "horrification"). In predicting the rating of a hotel review, we might find the following two aggregated features more relevant:

$$\widetilde{\boldsymbol{X}}_{\text{worry}} = \boldsymbol{X}_{\text{worrying}} + \boldsymbol{X}_{\text{depressing}} + \cdots + \boldsymbol{X}_{\text{troubling}}$$

$$\widetilde{\boldsymbol{X}}_{\text{horrification}} = \boldsymbol{X}_{\text{horrid}} + \boldsymbol{X}_{\text{hideous}} + \cdots + \boldsymbol{X}_{\text{awful}}.$$

The distinction between "horrid" and "hideous" might not matter for predicting the hotel rating, whereas the distinction between a "worry"-related word versus a "horrification"-related word may be quite relevant. Thus, not only are these aggregated features less rare than the original features, but they may also be more relevant to the prediction task. A method that selects the aggregated feature $\widetilde{\boldsymbol{X}}_{\text{horrification}}$ thereby can incorporate the information conveyed in the use of "hideous" into the prediction task; this same method may be unable to otherwise determine the effect of "hideous" by itself since it is too rare.

Indeed, appropriate aggregation of rare features in certain situations can be key to attaining consistent estimation and support recovery. In Theorem 5, we consider a setting where all features are rare and a natural aggregation rule exists among the features. In that setting, we show that the lasso (Tibshirani, 1996) fails to attain high-probability support recovery (for all values of its tuning parameter), whereas an oracle-aggregator does attain this property. Theorem 5 demonstrates the value of proper aggregation for accurate feature selection when features are rare. This motivates the remainder of the chapter, in which we devise a strategy for determining an effective feature aggregation based on data. Our aggregation procedure makes use of side information about the features, which we find is available in many domains. In particular, we assume that a tree is available that represents the closeness of features. For example, Figure 3.1 shows a tree for the previous word example that is generated via hierarchical clustering over `word2vec` (Mikolov et al., 2013; Mikolov et al., 2013) embeddings learned from a different data source. The two contours enclose two subtrees resulting from a cut at their joint node. Aggregating the counts in these subtrees leads to the new features $\widetilde{\boldsymbol{X}}_{\text{worry}}$ and $\widetilde{\boldsymbol{X}}_{\text{horrification}}$ described above. We

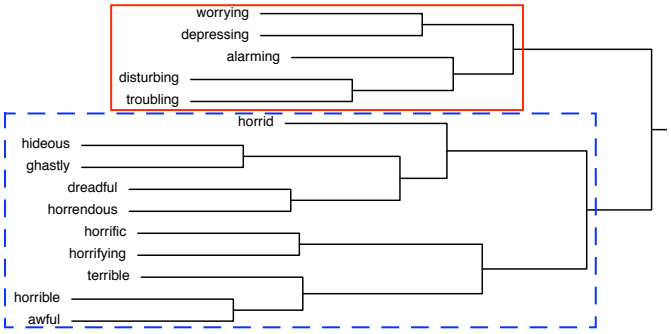


Figure 3.1: A tree that relates adjectives on its leaves

give more details of constructing such a tree in Section 3.3.1.

In Section 3.2, we motivate our work by providing theoretical results demonstrating the difficulty that OLS and the lasso have with rare features. We further show that correct aggregation of rare features leads to signed support recovery in a setting where the lasso is unable to attain this property. In Section 3.3, we introduce a tree-based parametrization strategy that translates the feature aggregation problem to a sparse modeling problem. Our main proposal is an estimator formulated as a solution to a convex optimization problem for which we derive an efficient algorithm. In Section 3.4, we prove a bound on the prediction error for our method. Finally, we demonstrate the empirical merits of the proposed framework through simulation (Section 3.5) and through the TripAdvisor prediction task (Section 3.6) described above. In simulation, we examine our method's robustness to misspecified side information. Quantitative and qualitative comparisons in the TripAdvisor example highlight the advantages of aggregating rare features.

**Notation:** Given a design matrix $\boldsymbol{X} \in \mathbb{R}^{n \times p}$, let $\boldsymbol{x}_i \in \mathbb{R}^p$ denote the feature vector of observation $i$ and $\boldsymbol{X}_j \in \mathbb{R}^n$ denote the $j$th feature, where $i = 1, \ldots, n$ and $j = 1, \ldots, p$. For a vector $\boldsymbol{\beta} \in \mathbb{R}^p$, let $\text{supp}(\boldsymbol{\beta}) \subseteq \{1, \ldots, p\}$ denote its support (i.e.,

the set of indices of nonzero elements). Let $\mathbb{S}_{\pm}(\boldsymbol{\beta}) := (sign(\beta_\ell))_{\ell=1,\ldots,p}$ encode the signed support of the vector $\boldsymbol{\beta}$. Let $\mathcal{T}$ be a $p$-leafed tree with root $r$, set of leaves $\mathcal{L}(\mathcal{T}) = \{1, \ldots, p\}$, and set of nodes $\mathcal{V}(\mathcal{T})$ of size $|\mathcal{T}|$. Let $\mathcal{T}_u$ be the subtree of $\mathcal{T}$ rooted by $u$ for $u \in \mathcal{V}(\mathcal{T})$. We follow the commonly-used notions of *child*, *parent*, *sibling*, *descendant*, and *ancestor* to describe the relationships between nodes of a tree. For a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, let $\|\boldsymbol{A}\|_1 = \max_{1 \le j \le n} \sum_{i=1}^{m} |a_{ij}|$ be the matrix-1 norm, $\|\boldsymbol{A}\|_\infty = \left\|\boldsymbol{A}^T\right\|_1$ be the matrix-$\infty$ norm, and (for a subset $B$ of $\{1, \ldots, n\}$) $\boldsymbol{A}_B \in \mathbb{R}^{m \times |B|}$ be the submatrix formed by removing the columns of $\boldsymbol{A}$ not in $B$. Let $S(\beta_\ell, \lambda) = sign(\beta_\ell) \cdot \max\{|\beta_\ell| - \lambda, 0\}$ be the soft-thresholding operator applied to $\beta_\ell \in \mathbb{R}$. We let $\boldsymbol{e}_j$ denote the vector having a one in the $j$th entry and zero elsewhere.

## 3.2 Rare Features and the Promise of Aggregation

### 3.2.1 The Difficulty Posed by Rare Features

Consider the linear model,

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}^* + \boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_n). \tag{3.1}$$

where $\boldsymbol{y} = (y_1, \ldots, y_n)^T \in \mathbb{R}^n$ is a response vector, $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ is a design matrix, $\boldsymbol{\beta}^*$ is a $p$-vector of parameters, and $\boldsymbol{\varepsilon} \in \mathbb{R}^n$ is a vector of independent Gaussian errors having variance $\sigma^2$. In this chapter, we focus on counts data, i.e., $\boldsymbol{X}_{ij}$ records the frequency of an event $j$ in observation $i$. In particular, we will assume throughout that $\boldsymbol{X}$ has non-negative elements.

The lasso (Tibshirani, 1996) is an estimator that performs variable selection,

making it well-suited to the $p \gg n$ setting:

$$\hat{\boldsymbol{\beta}}_{\lambda}^{lasso} \in \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min} \frac{1}{2n} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1. \tag{3.2}$$

When $\lambda = 0$, this coincides with the OLS estimator, which is uniquely defined when $n > p$ and $\boldsymbol{X}$ is full rank:

$$\hat{\boldsymbol{\beta}}^{\text{OLS}}(n) = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{y}.$$

To better understand the challenge posed by rare features, we begin by considering the effect of a single rare feature on OLS in the classical $p$-fixed, $n \to \infty$ regime. We take the $j$th feature to be a binary vector having $k$ nonzeros, where $k$ is a fixed value not depending on $n$. As $n$ increases, the proportion of nonzero elements, $k/n$, goes to 0. We show in Theorem 4 that $\hat{\beta}_j^{\text{OLS}}(n)$ does not converge in probability to $\beta_j^*$ with increasing sample size. This establishes that OLS is not a consistent estimator of $\boldsymbol{\beta}^*$ even in a $p$-fixed asymptotic regime.

**Theorem 4.** *Consider the linear model* (3.1) *with* $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ *having full column rank. Further suppose that* $\boldsymbol{X}_j$ *is a binary vector having (a constant) k nonzeros. It follows that there exists* $\eta > 0$ *for which*

$$\liminf_{n \to \infty} \mathbb{P}\left(\left|\hat{\beta}_j^{OLS}(n) - \beta_j^*\right| > \eta\right) > 0.$$

*Proof.* The result follows from taking $\liminf_{n \to \infty}$ of both sides of (B.1) in Appendix B.1 and observing that $2\Phi\left(-\eta k^{1/2}/\sigma\right)$ does not depend on $n$. □

The above result highlights the difficulty of estimating the coefficient of a rare feature. This suggests that even when rare features are not explicitly discarded, variable selection methods may fail to ever select them regardless of

their strength of association with the response. Other researchers have also acknowledged the difficulty posed by rare features in different scenarios. For example, in the context of hypothesis testing for high-dimensional sparse binary regression, Mukherjee et al. (2015) shows that when the design matrix is too sparse, any test has no power asymptotically, and signals cannot be detected regardless of their strength. Since the failure is caused by the sparsity of the features, it is therefore natural to ask if "densifying the features" in an appropriate way would fix the problem. As discussed above, aggregating the counts of related events may be a reasonable way to allow a method to make use of the information in rare features.

### 3.2.2 Aggregating Rare Features Can Help

Given $m$ subsets of $\{1, \ldots, p\}$, we can form $m$ aggregated features by summing within each subset. We can encode these subsets in a binary matrix $\boldsymbol{A} \in \{0,1\}^{p \times m}$ and form a new design matrix of aggregated features as $\widetilde{\boldsymbol{X}} = \boldsymbol{X}\boldsymbol{A}$. The columns of $\widetilde{\boldsymbol{X}}$ are also counts, but represent the frequency of $m$ different *unions* of the $p$ original events. For example, if the first subset is $\{1, 6, 8\}$, the first column of $\boldsymbol{A}$ would be $\boldsymbol{e}_1 + \boldsymbol{e}_6 + \boldsymbol{e}_8$ and the first aggregated feature would be $\widetilde{\boldsymbol{X}}_1 = \boldsymbol{X}_1 + \boldsymbol{X}_6 + \boldsymbol{X}_8$, recording the number of times any of the first, sixth, or eighth events occur. A linear model, $\widetilde{\boldsymbol{X}}\tilde{\boldsymbol{\beta}}$, based on the aggregated features can be equivalently expressed as a linear model, $\boldsymbol{X}\boldsymbol{\beta}$, in terms of the original features as long as $\boldsymbol{\beta}$ satisfies a set of linear constraints (ensuring that it is in the column space of $\boldsymbol{A}$):

$$\widetilde{\boldsymbol{X}}\tilde{\boldsymbol{\beta}} = (\boldsymbol{X}\boldsymbol{A})\tilde{\boldsymbol{\beta}} = \boldsymbol{X}(\boldsymbol{A}\tilde{\boldsymbol{\beta}}) = \boldsymbol{X}\boldsymbol{\beta}.$$

The vector $\boldsymbol{\beta}$ lies in the column space of $\boldsymbol{A}$ precisely when it is constant within each of the $m$ subsets. For example,

$$\textit{enforcing } \beta_1 = \beta_6 = \beta_8 \quad \Leftrightarrow \quad \textit{aggregating features: } \boldsymbol{X}_1\beta_1 + \boldsymbol{X}_6\beta_6 + \boldsymbol{X}_8\beta_8 = (\boldsymbol{X}_1 + \boldsymbol{X}_6 + \boldsymbol{X}_8)\beta_1 = \widetilde{\boldsymbol{X}}_1\tilde{\beta}_1. \tag{3.3}$$

In practice, determining how to aggregate features is a challenging problem, and our proposed strategy in Section 3.3 will use side information to guide this aggregation.

For now, to understand the potential gains achievable by aggregation, we consider an idealized case in which the correct aggregation of features is given to us by an oracle. In the next theorem, we construct a situation in which (a) the lasso on the original rare features is unable to correctly recover the support of $\boldsymbol{\beta}^*$ for any value of the tuning parameter $\lambda$, and (b) an oracle-aggregation of features makes it possible for the lasso to recover the support of $\boldsymbol{\beta}^*$. For simplicity, we take $\boldsymbol{X} = \boldsymbol{I}_n$, which corresponds to the case in which every feature has a single nonzero observation (and $n = p$). We take $\boldsymbol{\beta}^*$ to have $k$ blocks of size $n/k$, with entries that are constant within each block. The last block is all zeros and the minimal nonzero $|\beta_j^*|$, is restricted to lie within a range that expands with $n$ and shrinks with $k$. The oracle approach delivers to the lasso the $k$ aggregated features that match the structure in $\boldsymbol{\beta}^*$. These aggregated features have $n/k$ nonzeros, and thus are not rare features. Having peformed the lasso on these aggregated features, we then duplicate the $k$ elements, $n/k$ times per group, to get $\hat{\boldsymbol{\beta}}_\lambda^{oracle} \in \mathbb{R}^n$. The lasso with the oracle-aggregator is shown to achieve high-probability signed support recovery whereas the lasso on the original features fails to achieve this property for all values of the tuning parameter $\lambda$.

**Theorem 5.** *Consider the linear model* (3.1) *with* $\boldsymbol{X} = \boldsymbol{I}_n$ *and* $\boldsymbol{\beta}^* = \tilde{\boldsymbol{\beta}}^* \otimes \mathbf{1}_{n/k}$ *for* $\tilde{\boldsymbol{\beta}}^* = (\tilde{\beta}_1^*, \ldots, \tilde{\beta}_{k-1}^*, 0)$. *If* $\sigma\sqrt{\frac{4k\log(k^2n)}{n}} < \min_{i=1,\ldots,k-1} |\tilde{\beta}_i^*| \leq \sigma\sqrt{\frac{\log(2\tilde{c}(k-1)n/k)}{3}}$ *where*

$\tilde{c} = \frac{1}{3} e^{(\pi/2+2)^{-1}} \sqrt{\frac{1}{4} + \frac{1}{\pi}}$

*(a) The lasso fails to get high-probability signed support recovery:*

$$\limsup_{n\to\infty} \sup_{\lambda \geq 0} \mathbb{P}\left(\mathbb{S}_{\pm}(\hat{\boldsymbol{\beta}}_{\lambda}^{lasso}) = \mathbb{S}_{\pm}(\boldsymbol{\beta}^*)\right) \leq \frac{1}{e}.$$

*(b) The lasso with an oracle-aggregation of features succeeds in recovering the correct signed support for some $\lambda > 0$:*

$$\lim_{n\to\infty} \mathbb{P}\left(\mathbb{S}_{\pm}(\hat{\boldsymbol{\beta}}_{\lambda}^{oracle}) = \mathbb{S}_{\pm}(\boldsymbol{\beta}^*)\right) = 1.$$

*Proof.* These results correspond to Propositions 3 and 4 in Appendix B.2 and are proved there. □

## 3.3 Main Proposal: Tree-Guided Aggregation

In the previous section, we have seen the potential gains achievable through aggregating rare features. In this section, we propose a tree-guided method for aggregating and selecting rare features. We discuss this tree in Section 3.3.1, introduce a tree-based parametrization strategy in Section 3.3.2, and propose a new estimator in Section 3.3.3.

### 3.3.1 A Tree to Guide Aggregation

To form aggregated variables, it is infeasible to consider all possible partitions of the features $\{1, \ldots, p\}$. Rather, we will consider a tree $\mathcal{T}$ with leaves $1, \ldots, p$ and restrict ourselves to partitions that can be expressed as a collection of branches

of $\mathcal{T}$ (see, e.g., Figure 3.1). We sum features within a branch to form our new aggregated features.

We would like to aggregate features that are related, and thus we would like to have $\mathcal{T}$ encode feature similarity information. Such information about the features comes from prior knowledge and/or data sources external to the current regression problem (i.e., not from $y$ and $X$). For example, for microbiome data, $\mathcal{T}$ could be the phylogenetic tree encoding evolutionary relationships among the OTUs (e.g., Matsen et al. 2010; Tang et al. 2016; Wang and Zhao 2017a) or the co-occurrence of OTUs from past data sets. When features correspond to words, closeness in meaning can be used to form $\mathcal{T}$ (e.g., in Section 3.6, we perform hierarchical clustering on word embeddings that were learned from an enormous corpus).

In (3.3), we demonstrated how aggregating a set of features is equivalent to setting these features' coefficients to be equal. To perform tree-guided aggregation, we therefore associate a coefficient $\beta_j$ with each leaf of $\mathcal{T}$ and "fuse" (i.e., set equal to each other) any coefficients within a branch that we wish to aggregate.

### 3.3.2 A Tree-Based Parametrization

In order to fuse $\beta_j$'s within a branch, we adopt a tree-based parametrization by assigning a parameter $\gamma_u$ to each node $u$ in $\mathcal{T}$ (this includes both leaves and interior nodes). The left panel of Figure 3.2 gives an example. Let $ancestor(j) \cup \{j\}$ be the set of nodes in the path from the root of $\mathcal{T}$ to the $j$th feature, which is

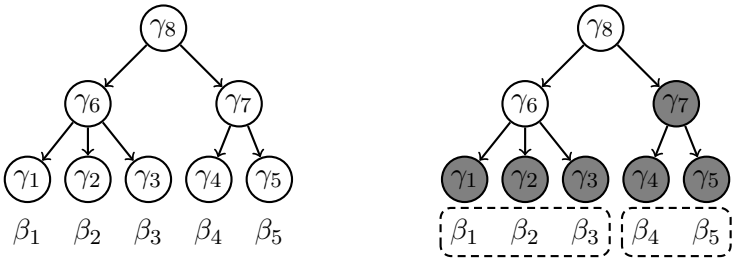


Figure 3.2: (Left) An example of $\boldsymbol{\beta} \in \mathbb{R}^5$ and $\mathcal{T}$ that relates the corresponding five features. By (3.4), we have $\beta_i = \gamma_i + \gamma_6 + \gamma_8$ for $i = 1, 2, 3$ and $\beta_j = \gamma_j + \gamma_7 + \gamma_8$ for $j = 4, 5$. (Right) By zeroing out the $\gamma_i$'s in the gray nodes, we aggregate $\beta$ into two groups indicated by the dashed contours: $\beta_1 = \beta_2 = \beta_3 = \gamma_6 + \gamma_8$ and $\beta_4 = \beta_5 = \gamma_8$. Counts data are aggregated for features sharing the same coefficient: $\boldsymbol{X\beta} = (\boldsymbol{X}_1 + \boldsymbol{X}_2 + \boldsymbol{X}_3)\beta_1 + (\boldsymbol{X}_4 + \boldsymbol{X}_5)\beta_4$.

associated with the $j$th leaf. We express $\beta_j$ as the sum of all the $\gamma_u$'s on the path:

$$\beta_j = \sum_{u \in ancestor(j) \cup \{j\}} \gamma_u. \tag{3.4}$$

This can be written more compactly as $\boldsymbol{\beta} = \boldsymbol{A\gamma}$, where $\boldsymbol{A} \in \{0, 1\}^{p \times |\mathcal{T}|}$ is a binary matrix with $A_{jk} := 1_{\{u_k \in ancestor(j) \cup \{j\}\}} = 1_{\{j \in descendant(u_k) \cup \{u_k\}\}}$. The descendants of each node $u$ define a branch $\mathcal{T}_u$, and zeroing out $\gamma_v$'s for all $v \in descendant(u)$ fuses the coefficients in this branch, i.e., $\{\beta_j : j \in \mathcal{L}(\mathcal{T}_u)\}$. Thus, $\gamma_{descendant(u)} = 0$ is equivalent to aggregating the features $\boldsymbol{X}_j$ with $j \in \mathcal{L}(\mathcal{T}_u)$ (see the right panel of Figure 3.2).

Another way of viewing this parametrization's merging of branches is by expressing $\boldsymbol{X\beta} = \boldsymbol{XA\gamma}$, where $(XA)_{ik} = \sum_{j=1}^{p} X_{ij} A_{jk} = \sum_{j: j \in descendant(u_k) \cup \{u_k\}} X_{ij}$ aggregates counts over all the descendant features of node $u_k$. By aggregating nearby features, we allow rare features to borrow strength from their neighbors, allowing us to estimate shared coefficient values that would otherwise be too difficult to estimate. In the next section, we describe an optimization problem that uses the $\boldsymbol{\gamma}$ parametrization to simultaneously perform feature aggregation and selection.

### 3.3.3 The Optimization Problem

Our proposed estimator $\hat{\beta}$ is the solution to the following convex optimization problem:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p,\boldsymbol{\gamma}\in\mathbb{R}^{|\mathcal{T}|}} \left\{ \frac{1}{2n} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda \left(\alpha \|\boldsymbol{\gamma}_{-r}\|_1 + (1-\alpha)\|\boldsymbol{\beta}\|_1\right) \quad \text{s.t. } \boldsymbol{\beta} = \boldsymbol{A}\boldsymbol{\gamma} \right\}. \tag{3.5}$$

We apply an $\ell_1$ penalty on non-root $\gamma_u$'s to induce sparsity in $\hat{\boldsymbol{\gamma}}$, which in turn induces fusion of the coefficients in $\hat{\boldsymbol{\beta}}$. In the high-dimensional setting, sparsity in feature coefficients is also desirable. Therefore, we apply an $\ell_1$ penalty on $\boldsymbol{\beta}$ as well. The tuning parameter $\lambda$ controls the overall penalization level while $\alpha$ determines the trade-off between the two types of regularization: fusion and sparsity. In practice, both $\lambda$ and $\alpha$ are determined via cross validation.

When $\alpha = 0$, (3.5) reduces to a lasso problem in $\boldsymbol{\beta}$; when $\alpha = 1$, (3.5) reduces to a lasso problem in $\boldsymbol{\gamma}$. Both extreme cases can be efficiently solved with a lasso solver such as `glmnet` (Friedman et al., 2010). For $\alpha \in (0, 1)$, (3.5) is a generalized lasso problem (Tibshirani and Taylor, 2011) in $\boldsymbol{\gamma}$, and can be solved in principle using preexisting solvers (e.g., Arnold and Tibshirani 2014). However, better computational performance, in particular in high-dimensional settings, can be attained using an algorithm specially tailored to our problem. We write (3.5) as a *global consensus problem* and solve this using alternating direction method of multipliers (ADMM, Boyd et al. (2011)). The consensus problem introduces additional copies of $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, which decouples the various parts of the problem, leading to efficient ADMM updates:

$$\min_{\substack{\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{\beta}^{(3)},\boldsymbol{\beta}\in\mathbb{R}^p \\ \text{and } \boldsymbol{\gamma}^{(1)},\boldsymbol{\gamma}^{(2)},\boldsymbol{\gamma}\in\mathbb{R}^{|\mathcal{T}|}}} \left\{ \frac{1}{2n} \left\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}^{(1)}\right\|_2^2 + \lambda\alpha\|\boldsymbol{\gamma}_{-r}^{(1)}\|_1 + \lambda(1-\alpha)\|\boldsymbol{\beta}^{(2)}\|_1 \right\} \tag{3.6}$$

$$\text{s.t. } \boldsymbol{\beta}^{(3)} = \boldsymbol{A}\boldsymbol{\gamma}^{(2)}, \boldsymbol{\beta} = \boldsymbol{\beta}^{(1)} = \boldsymbol{\beta}^{(2)} = \boldsymbol{\beta}^{(3)} \text{ and } \boldsymbol{\gamma} = \boldsymbol{\gamma}^{(1)} = \boldsymbol{\gamma}^{(2)}.$$

In particular, our ADMM approach requires performing a singular value decomposition (SVD) on $\boldsymbol{X}$, an SVD on $(\boldsymbol{I}_p : -\boldsymbol{A})$ (these are reused for all $\lambda$ and $\alpha$), and then applying matrix multiplies and soft-thresholdings until convergence. Let $\boldsymbol{X} = \text{SVD}_{\text{compact}}(\widetilde{\boldsymbol{U}}, \widetilde{\boldsymbol{D}}, \widetilde{\boldsymbol{V}})$ be the *compact* singular value decomposition of $\boldsymbol{X}$, where $\widetilde{\boldsymbol{D}} \in \mathbb{R}^{\min(n,p)\times\min(n,p)}$ is a diagonal matrix with non-zero singular values on the diagonal, and $\widetilde{\boldsymbol{U}} \in \mathbb{R}^{n\times\min(n,p)}$ and $\widetilde{\boldsymbol{V}} \in \mathbb{R}^{p\times\min(n,p)}$ contain the left and right singular vectors in columns corresponding to non-zero singular values, respectively. Similarly, we have $(\boldsymbol{I}_p : -\boldsymbol{A}) = \text{SVD}_{\text{compact}}(\cdot, \cdot, \widetilde{\boldsymbol{Q}})$ where $\widetilde{\boldsymbol{Q}} \in \mathbb{R}^{(p+|\mathcal{T}|):p}$ contains $p$ right singular vectors correponding to non-zero singular values. See Algorithm 5 for details. Appendix B.3.1 provides a derivation of Algorithm 5 and Appendix B.3.2 discusses a slight modification for including an intercept, which is desirable in practice.

We conclude this section by making connections to some related work. The idea of using a tree as auxiliary information for achieving different tasks appears, for example, in the genomics literature. Wang and Zhao (2017b) introduce a penalized regression method with high-dimensional and compositional covariates that uses a phylogenetic tree; however, their goal and use of the tree is fundamentally different from ours. Their tree-based penalty maintains a zero-sum constraint for coefficients in order to perform subcomposition selection rather than feature aggregation. (In fact, in their application to the human gut microbiome study, Wu et al. 2011, before applying their procedure they begin by manually aggregating 17,000 OTUs to 62 in order to reduce the sparsity of the data.) Guinot et al. (2017) considers a similar idea of aggregating genomic features with the help of a hierarchical clustering tree; however, their tree is learned from the design matrix and the prediction task is only used to determine the level of tree cut, whereas our method uses the response to flexibly

**Algorithm 5** Consensus ADMM for Solving Problem (3.5)

**Input:** $\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{A}, n, p, |\mathcal{T}|, \lambda, \alpha, \rho, \epsilon^{abs}, \epsilon^{rel}$, `maxite`.

$\boldsymbol{X} = \text{SVD}_{\text{compact}}(\cdot, \widetilde{\boldsymbol{D}}, \widetilde{\boldsymbol{V}})$

$(\boldsymbol{I}_p : -\boldsymbol{A}) = \text{SVD}_{\text{compact}}(\cdot, \cdot, \widetilde{\boldsymbol{Q}})$

$\boldsymbol{\beta}^0 \leftarrow \boldsymbol{\beta}^{(i)0} \leftarrow \boldsymbol{v}^{(i)0} \leftarrow \boldsymbol{0} \in \mathbb{R}^p \quad \forall i = 1, 2, 3$

$\boldsymbol{\gamma}^0 \leftarrow \boldsymbol{\gamma}^{(j)0} \leftarrow \boldsymbol{u}^{(j)0} \leftarrow \boldsymbol{0} \in \mathbb{R}^{|\mathcal{T}|} \quad \forall j = 1, 2$

`continue` $\leftarrow$ `true`

$k \leftarrow 0$

**while** $k <$ `maxite` **and** `continue` **do**

  $k \leftarrow k + 1$

  $\boldsymbol{\beta}^{(1)k} \leftarrow \left[\widetilde{\boldsymbol{V}} diag\left(\frac{1}{[\widetilde{\boldsymbol{D}}^T\widetilde{\boldsymbol{D}}]_{ii}+n\rho}\right)\widetilde{\boldsymbol{V}}^T + \frac{1}{n\rho}(\boldsymbol{I}_p - \widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{V}}^T)\right]\left(\boldsymbol{X}^T\boldsymbol{y} + n\rho\boldsymbol{\beta}^{k-1} - n\boldsymbol{v}^{(1)k-1}\right)$

  $\beta_\ell^{(2)k} \leftarrow S\left(\beta_\ell^{k-1} - \frac{1}{\rho}v_\ell^{(2)k-1}, \frac{\lambda(1-\alpha)}{\rho}\right) \quad \forall \ell = 1, \ldots, p$

  $\gamma_\ell^{(1)k} \leftarrow \begin{cases} S\left(\gamma_\ell^{k-1} - \frac{1}{\rho}u_\ell^{(1)k-1}, \frac{\lambda\alpha}{\rho}\right) & \text{if } \ell \in \{1, \ldots, |\mathcal{T}|\}\backslash\{r\} \\ \gamma_\ell^{k-1} - \frac{1}{\rho}u_\ell^{(1)k-1} & \text{if } \ell = r \end{cases}$

  $\begin{pmatrix}\boldsymbol{\beta}^{(3)k} \\ \boldsymbol{\gamma}^{(2)k}\end{pmatrix} \leftarrow \left(\boldsymbol{I}_{p+|\mathcal{T}|} - \widetilde{\boldsymbol{Q}}\widetilde{\boldsymbol{Q}}^T\right)\left[\begin{pmatrix}\boldsymbol{\beta}^{k-1} \\ \boldsymbol{\gamma}^{k-1}\end{pmatrix} - \frac{1}{\rho}\begin{pmatrix}\boldsymbol{v}^{(3)k-1} \\ \boldsymbol{u}^{(2)k-1}\end{pmatrix}\right]$

  $\boldsymbol{\beta}^k \leftarrow (\boldsymbol{\beta}^{(1)k} + \boldsymbol{\beta}^{(2)k} + \boldsymbol{\beta}^{(3)k})/3$

  $\boldsymbol{\gamma}^k \leftarrow (\boldsymbol{\gamma}^{(1)k} + \boldsymbol{\gamma}^{(2)k})/2$

  $\boldsymbol{v}^{(i)k} \leftarrow \boldsymbol{v}^{(i)k-1} + \rho(\boldsymbol{\beta}^{(i)k} - \boldsymbol{\beta}^k) \quad \forall i = 1, 2, 3$

  $\boldsymbol{u}^{(j)k} \leftarrow \boldsymbol{u}^{(j)k-1} + \rho(\boldsymbol{\gamma}^{(j)k} - \boldsymbol{\gamma}^k) \quad \forall j = 1, 2.$

  **if** $\sqrt{\sum_{i=1}^3 \|\boldsymbol{\beta}^{(i)k} - \boldsymbol{\beta}^k\|_2^2 + \sum_{j=1}^2 \|\boldsymbol{\gamma}^{(j)k} - \boldsymbol{\gamma}^k\|_2^2} \leq \epsilon^{abs}\sqrt{3p + 2|\mathcal{T}|} + \epsilon^{rel}\max\left\{\sqrt{\sum_{i=1}^3 \|\boldsymbol{\beta}^{(i)k}\|_2^2 + \sum_{j=1}^2 \|\boldsymbol{\gamma}^{(j)k}\|_2^2}, \sqrt{3\|\boldsymbol{\beta}^k\|_2^2 + 2\|\boldsymbol{\gamma}^k\|_2^2}\right\}$
  **and** $\rho\sqrt{3\|\boldsymbol{\beta}^k - \boldsymbol{\beta}^{k-1}\|_2^2 + 2\|\boldsymbol{\gamma}^k - \boldsymbol{\gamma}^{k-1}\|_2^2} \leq \epsilon^{abs}\sqrt{3p + 2|\mathcal{T}|} + \epsilon^{rel}\sqrt{\sum_{i=1}^3 \|\boldsymbol{v}^{(i)k}\|_2^2 + \sum_{j=1}^2 \|\boldsymbol{u}^{(j)k}\|_2^2}$ **then**

    `continue` $\leftarrow$ `false`

  **end if**

**end while**

**Output:** $\boldsymbol{\beta}^k, \boldsymbol{\gamma}^k$

choose differing aggregation levels across the tree. We consider a strategy similar to theirs, which we call `L1-ag-h` in the empirical comparisons. Finally, Kim et al. (2012) propose a tree-guided group lasso approach in the context of multi-response regression. In their context, the tree relates the different responses and is used to borrow strength across related prediction tasks.

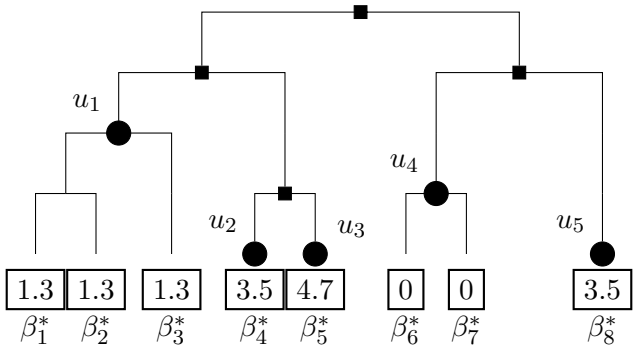


Figure 3.3: In the above tree, $B^* = \{u_1, u_2, u_3, u_4, y_5\}$ has its nodes labeled with black circles.

## 3.4 Statistical Theory

In this section, we study the prediction consistency of our method. Since $\mathcal{T}$ encodes feature similarity information, throughout the section we require $\mathcal{T}$ to be a "full" tree such that each node is either a leaf or possesses at least two child nodes. We begin with some definitions.

**Definition 1.** We say that $B \subseteq \mathcal{V}(\mathcal{T})$ is an *aggregating set* with respect to $\mathcal{T}$ if $\{\mathcal{L}(\mathcal{T}_u) : u \in B\}$ forms a partition of $\mathcal{L}(\mathcal{T})$.

The black circles in Figure 3.3 form an aggregating set since their branches' leaves are a partition of $\{1, \ldots, 8\}$. We would like to refer to "the true aggregating set $B^*$ with respect to $\mathcal{T}$" and, to do so, we must first establish that there exists a unique coarsest aggregating set corresponding to a vector $\boldsymbol{\beta}^*$.

**Lemma 6.** *For any $\boldsymbol{\beta}^* \in \mathbb{R}^p$, there exists a unique coarsest aggregating set $B^* := B(\boldsymbol{\beta}^*, \mathcal{T}) \subseteq \mathcal{V}(\mathcal{T})$ (hereafter "the aggregating set") with respect to the tree $\mathcal{T}$ such that (a) $\beta_j^* = \beta_k^*$ for $j, k \in \mathcal{L}(\mathcal{T}_u)$ $\forall u \in B^*$, (b) $|\beta_j^* - \beta_k^*| > 0$ for $j \in \mathcal{L}(\mathcal{T}_u)$ and $k \in \mathcal{L}(\mathcal{T}_v)$ for siblings $u, v \in B^*$.*

The lemma (proved in Appendix B.4) defines $B^*$ as the aggregating set such

that further merging of siblings would mean that $\beta^*$ is not constant within each subset of the partition.

**Definition 2.** Given the triplet $(\mathcal{T}, \boldsymbol{\beta}^*, \boldsymbol{X})$, we define (a) $\widetilde{\boldsymbol{X}} = \boldsymbol{X}\boldsymbol{A}_{B^*} \in \mathbb{R}^{n \times |B^*|}$ to be the *design matrix of aggregated features*, which uses $B^* = B(\boldsymbol{\beta}^*, \mathcal{T})$ as the aggregating set, and (b) $\tilde{\boldsymbol{\beta}}^* \in \mathbb{R}^{|B^*|}$ to be the coefficient vector using these aggregated features: $\boldsymbol{\beta}^* = \boldsymbol{A}_{B^*}\tilde{\boldsymbol{\beta}}^*$.

We are now ready to provide a bound on the prediction error of our estimator, which is proved in Appendix B.5.

**Theorem 6** (Prediction Error Bound)**.** *Assume $\boldsymbol{X}$ has been scaled so that $\left\|\boldsymbol{X}\mathbf{1}_p\right\|_2^2 = n$. If we take $\lambda \geq 8\sigma\sqrt{\frac{\log 2p}{n}}$ and $0 \leq \alpha \leq (1 + p^{-1})^{-1}$, then with probability at least $1 - p^{-1}$,*

$$\frac{1}{n}\left\|\boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{X}\boldsymbol{\beta}^*\right\|_2^2 \leq 3\lambda\left(\alpha\left\|\tilde{\boldsymbol{\beta}}^*\right\|_1 + (1-\alpha)\left\|\boldsymbol{\beta}^*\right\|_1\right).$$

The above theorem is an example of a slow-rate bound, which is notable for the fact that it places no assumptions on the design matrix $\boldsymbol{X}$ (Dalalyan et al., 2017). (The condition that $\left\|\boldsymbol{X}\mathbf{1}_p\right\|_2^2 = n$ is easily satisfied by appropriate scaling of $\boldsymbol{X}$). The bound depends on $\|\boldsymbol{\beta}^*\|_1$ and $\|\tilde{\boldsymbol{\beta}}^*\|_1$, which heuristically can be thought of as measuring the number of original and aggregated features that are relevant for making predictions. The following corollary facilitates the interpretation of the prediction bound and demonstrates the effectiveness of our method given a good choice for $\alpha$.

**Corollary 1.** *Assume that $\|\beta^*\|_\infty \leq M$ and $\boldsymbol{X}$ has been scaled so that $\left\|\boldsymbol{X}\mathbf{1}_p\right\|_2^2 = n$. Taking $\lambda = 8\sigma\sqrt{\frac{\log 2p}{n}}$ and $0 \leq \alpha \leq (1 + p^{-1})^{-1}$, we have, with probability at least $1 - p^{-1}$, that*

$$\frac{1}{n}\left\|\boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{X}\boldsymbol{\beta}^*\right\|_2^2 \leq 24\sigma M\sqrt{\frac{\log 2p}{n}}\left(\alpha|B^*| + (1-\alpha)|\mathcal{A}^*|\right),$$

*where $\mathcal{A}^*$ is the support of $\boldsymbol{\beta}^*$. Furthermore, with $\alpha = \frac{|\mathcal{A}^*|}{|\mathcal{A}^*|+|B^*|}$,*

$$\frac{1}{n}\left\|\boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{X}\boldsymbol{\beta}^*\right\|_2^2 \leq 48\sigma M\sqrt{\frac{\log 2p}{n}} \cdot \min\left(|\mathcal{A}^*|, |B^*|\right)$$

*holds with probability at least $1 - p^{-1}$.*

*Proof.* See Appendix B.6. □

The first statement in Corollary 1 establishes that our estimator can make excellent predictions even when $p \gg n$ as long as $(\log p)/n \cdot \min(|\mathcal{A}^*|^2, |B^*|^2) \to 0$. The quantity $|\mathcal{A}^*|$ is the traditional notion of sparsity appearing throughout the high-dimensional statistics literature. The quantity $|B^*|$ is specific to our framework: it depends both on $\boldsymbol{\beta}^*$ and on the tree $\mathcal{T}$ that guides the aggregation. The second statement in Corollary 1 exhibits the effectiveness of our method. With a properly chosen $\alpha$, we can do well unless both $|\mathcal{A}^*|$ and $|B^*|$ are large. This is notable since one can construct extreme examples of $\boldsymbol{\beta}^*$ and $\mathcal{T}$ in which $|\mathcal{A}^*| = p$ and $|B^*| = 1$ and in which $|\mathcal{A}^*| = 1$ and $|B^*| = p$. That is, our method can do well even when there is no sparsity (as long as there is aggregation) and when there is no aggregation (as long as there is sparsity).

## 3.5 Simulation Study

For $k \in \{5, 10, 15, 20, 25, 30\}$, we generate $p$ data points in a $(k-1)$-dimensional latent space, in which $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k \in \mathbb{R}^{k-1}$ are taken to be equidistant and as vertices of the unit simplex. We generate $p/k$ latent vectors independently from $N(\boldsymbol{\mu}_i, 0.1^2\boldsymbol{I}_{k-1})$ for $i \in \{1, \ldots, k\}$. We form a tree $\mathcal{T}$ by performing hierarchical clustering (using `hclust` in R Core Team (2016)) on the $p$ vectors, before cutting the

tree into $k$ subtrees (the roots of which form $B^*$). We form $\boldsymbol{A}$ corresponding to this tree and generate $\boldsymbol{\beta}^* = \boldsymbol{A}_{B^*}\tilde{\boldsymbol{\beta}}^*$. The first $k \cdot s$ elements of $\tilde{\boldsymbol{\beta}}^* \in \mathbb{R}^k$ are 0, and the remaining elements are drawn independently from a $N(0, 4)$ distribution. The design matrix $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ is simulated from a Poisson(0.1) distribution. The response $\boldsymbol{y} \in \mathbb{R}^n$ is simulated from (3.1) with $\sigma = \|\boldsymbol{X}\boldsymbol{\beta}^*\|_2/(5n)$. For every method under consideration, we average its performance over 100 repetitions in all the following simulations.

We consider two scenarios, one low-dimensional ($n = 500$, $p = 100$, $s = 0$) and the other high-dimensional ($n = 100$, $p = 200$, $s = 0.2$). We apply our method with the true $\mathcal{T}$ and vary the tuning parameters $(\alpha, \lambda)$ along an 8-by-50 grid of values. We compare our method to *oracle least squares*, in which we perform least squares on $[\boldsymbol{X}\boldsymbol{A}_{B^*}]_{(k\cdot s+1):k}$, the correctly aggregated features having non-zero $\tilde{\boldsymbol{\beta}}^*$. Oracle least squares represents the best possible performance of any method that attempts to aggregate and select features. In the low-dimensional scenario, we include least squares on the original design matrix $\boldsymbol{X}$, and in the high-dimensional scenario, we include the lasso and ridge regression, which are each computed across a grid of 50 values of the tuning parameter.

To understand the best performance attainable by each method, we measure the *best mean-squared estimation error*, i.e., $\min_\Lambda \|\hat{\boldsymbol{\beta}}(\Lambda) - \boldsymbol{\beta}^*\|_2^2/p$, where "best" is with respect to each method's tuning parameter(s) $\Lambda$. The left and the middle panels of Figures 3.4 shows the performance of the methods in the low-dimensional and high-dimensional scenarios, respectively. Given that our method includes least squares and the lasso as special cases, it is no surprise that our methods have better attainable performance than those methods. These results indicate that our method performs similarly to the oracle when the true

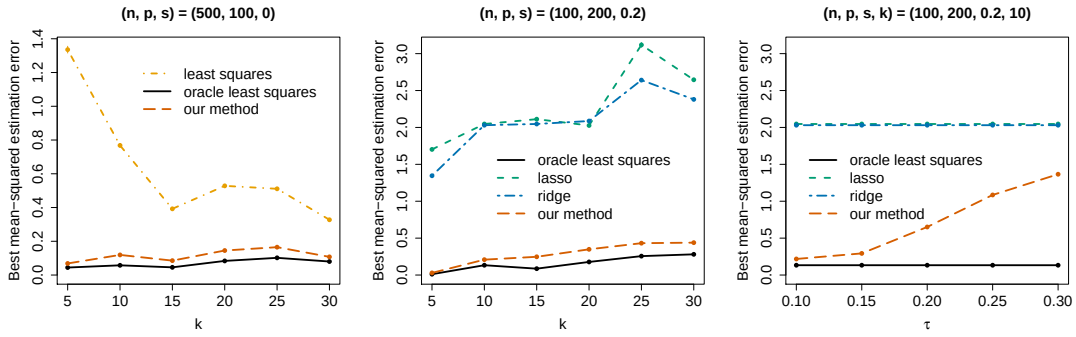


Figure 3.4: (Left and Middle) two scenarios for varying $k$: $\min_\Lambda \|\hat{\boldsymbol{\beta}}(\Lambda) - \boldsymbol{\beta}^*\|_2^2/p$ versus $k$ for $(n, p, s) = (500, 100, 0)$ and $(n, p, s) = (100, 200, 0.2)$. (Right) degredation of our method with distorted trees: $\min_\Lambda \|\boldsymbol{X}\hat{\boldsymbol{\beta}}(\Lambda) - \boldsymbol{X}\boldsymbol{\beta}^*\|_2^2/n$ versus $\tau$ for $(n, p, s, k) = (100, 200, 0.2, 10)$.

number of aggregated features, $k$, is small and degrades as this quantity increases.

Clearly, the performance of our method will depend on the quality of the tree being used. In the previous simulations we provided our method with a tree that is perfectly compatible with the true aggregating set. In practice, the tree used may be only an approximate representation of how features should be aggregated. We therefore study the sensitivity of our method to misspecification of the tree. We return to the high-dimensional setting above with $k = 10$, and we generate a sequence of trees that are increasingly distorted representations of how the data should in fact be aggregated.

We begin with a true aggregation of the features into $k$ groups, each of size $p/k$. In each repetition of the simulation, we generate a (random) tree $\mathcal{T}$ by performing hierarchical clustering on $p$ random vectors generated similarly as above: for each group $i \in \{1, \ldots, k\}$, we associate a cluster center $\boldsymbol{\mu}_i \in \mathbb{R}^{k-1}$ and generate $p/k$ latent vectors independently from $N(\boldsymbol{\mu}_i, \tau^2 \boldsymbol{I}_{k-1})$. We control the degradation level of the tree by varying the value of $\tau$. When $\tau$ is small, the latent vectors will be well-separated by group so that the tree will have an

aggregating set that matches the true aggregation structure (with high probability). As $\tau \in \{0.1, 0.15, 0.2, 0.25, 0.3\}$ increases, the information provided by the tree becomes increasingly weak. The right panel of Figure 3.4 shows the degradation of our method as $\tau$ increases.

## 3.6 Application to Hotel Reviews

Wang et al. (2010) crawled TripAdvisor.com to form a dataset[2] of 235,793 reviews and ratings of 1,850 hotels by users between February 14, 2009 and March 15, 2009. While there are several kinds of ratings, we focus on a user's overall rating of the hotel (on a 1 to 5 scale), which we take as our response. We form a document-term matrix $\boldsymbol{X}$ in which $\boldsymbol{X}_{ij}$ is the number of times the $i$th review uses the $j$th adjective.

We begin by converting words to lower case and keeping only adjectives (as determined by WordNet Fellbaum 1998; Wallace 2007; Feinerer and Hornik 2016). After removing reviews with missing ratings, we are left with 209,987 reviews and 7,787 distinct adjectives. The left panel of Figure 3.5 shows the distribution of ratings in the data: nearly three quarters of all ratings are above 3 stars. The extremely right-skewed distribution in the right panel of Figure 3.5 shows that all but a small number of adjectives are highly rare (e.g., over 90% of adjectives are used in fewer than 0.5% of reviews).

Rather than discard this large number of rare adjectives, our method aims to make productive use of these by leveraging side information about the relationship between adjectives. We construct a tree capturing adjective similar-

[2]Data source: `http://times.cs.uiuc.edu/~wang296/Data/`

| Rating | Proportion |
|---|---|
| 1 | 0.066 |
| 2 | 0.085 |
| 3 | 0.105 |
| 4 | 0.308 |
| 5 | 0.436 |

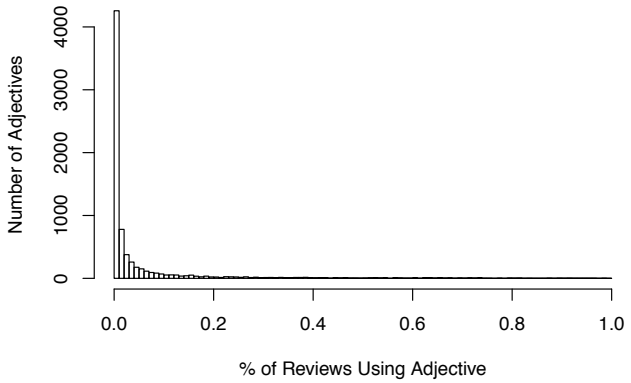


Figure 3.5: (Left) distribution of TripAdvisor ratings. (Right) only 414 adjectives appear in more than 1% of reviews; the histogram gives the distribution of usage-percentages for those adjectives appearing in fewer than 1% of reviews.

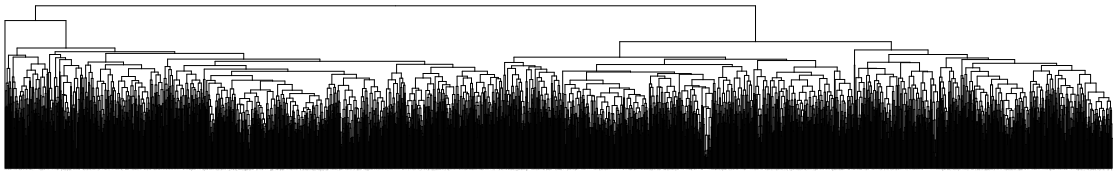

Figure 3.6: Tree $\mathcal{T}$ over 2,397 adjectives: the left subtree is for adjectives with negative sentiment and the right subtree is for adjectives with positive sentiment.

ity as follows. We start with word embeddings[3] in a 100-dimensional space that were pre-trained by *GloVe* (Pennington et al., 2014) on the Gigaword5 and Wikipedia2014 corpora. We also obtain a list of adjectives, which the NRC Emotion Lexicon labels as having either positive or negative sentiments (Mohammad and Turney, 2013). We use five nearest neighbors classification within the 100-dimensional space of word embeddings to assign labels to the 5,795 adjectives that have not been labeled in the NRC Emotion Lexicon. This sentiment separation determines the two main branches of the tree $\mathcal{T}$. Within each branch, we perform hierarchical clustering of the word embedding vectors. Figure 3.6 depicts such a tree with 2,397 adjectives (as leaves).

We compare our method to four other approaches, meant to represent vari-

[3]Data source: `http://nlp.stanford.edu/data/glove.6B.zip`

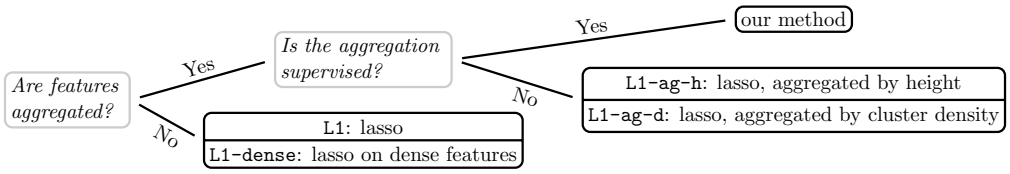


Figure 3.7: A comparison between our method and four other methods

ations of how the lasso is typically applied when rare features are present (see Figure 3.7 for a schematic). The most common and straightforward approach, which we refer to as `L1`, is to simply apply the lasso on the features without making any adjustment for rare features. A second approach, which we refer to as `L1-dense`, applies the lasso after first discarding any adjectives that are in fewer than 0.5% of reviews. The third and fourth approaches apply the lasso with features aggregated according to the tree in an unsupervised manner. The third approach, `L1-ag-h`, aggregates features that are in the same cluster after cutting the dendrogram at a certain height. In addition to the lasso tuning parameter, the height at which we cut the tree is a second tuning parameter (chosen along an equally-spaced grid of ten values). The fourth approach, `L1-ag-d`, performs merges in a bottom-up fashion along the tree until all aggregated features have density above some threshold. This threshold is an additional tuning parameter (chosen along an equally spaced grid of ten values between 0.001 and 0.1).

We hold out 40,000 ratings and reviews as a test set. To observe the performance of these methods over a range of training set sizes, we consider a nested sequence of training sets, ranging from 1% to 100% of the reviews not included in the test set. For all methods, we use five-fold cross validation to select tuning parameters and threshold all predicted ratings to be within the interval $[1, 5]$. Table 3.1 displays the mean squared prediction error (MSPE) on the test set for

Table 3.1: Performance of five methods on the held-out test set: `L1` is the lasso; `L1-dense` is the lasso on only dense features; `L1-ag-h` is the lasso with features aggregated based on height; and `L1-ag-d` is the lasso with features aggregated based on density level.

| | | | | Mean Squared Prediction Error | | | | |
|---|---|---|---|---|---|---|---|---|
| prop. | $n$ | $p$ | $n/p$ | our method | L1 | L1-dense | L1-ag-h | L1-ag-d |
| 1% | 1,700 | 2,397 | 0.71 | **0.870** | 0.894 | 0.895 | 0.882 | 0.971 |
| 5% | 8,499 | 3,962 | 2.15 | **0.783** | 0.790 | 0.805 | 0.785 | 0.899 |
| 10% | 16,999 | 4,786 | 3.55 | **0.758** | 0.764 | 0.788 | 0.764 | 0.902 |
| 20% | 33,997 | 5,621 | 6.05 | **0.742** | 0.749 | 0.773 | 0.747 | 1.173 |
| 40% | 67,995 | 6,472 | 10.51 | **0.739** | 0.740 | 0.768 | 0.742 | 1.108 |
| 60% | 101,992 | 6,962 | 14.65 | **0.733** | 0.736 | 0.769 | 0.734 | 1.155 |
| 80% | 135,990 | 7,294 | 18.64 | **0.733** | **0.733** | 0.765 | 0.734 | 0.886 |
| 100% | 169,987 | 7,573 | 22.45 | **0.729** | 0.731 | 0.765 | 0.731 | 0.956 |

each method and training set size.

As the size of the training set increases, all methods except for the lasso with aggregation based on density (`L1-ag-d`) achieve lower MSPE. Among the four lasso-related methods, `L1` and `L1-ag-h` outperform the other two. As the training set size $n$ increases, the number of features $p$ also increase but at a relatively slower rate. We notice that when $n/p$ is less than 10.51, our method outperforms the other four lasso-related methods. As $n/p$ increases beyond 10.51, i.e., in the statistically easier regimes, `L1` and `L1-ag-h` attain performance comparable to our method.

To better understand the difference between our method and the lasso, we color the branches of the tree generated in the $n = 1,700$ and $p = 2,397$ case (i.e., proportion is 1%) according to the sign and magnitude of $\hat{\boldsymbol{\beta}}$ for the two methods. The lower tree in Figure 3.8 corresponds to our method and has many nearby branches sharing the same color in (red or blue), indicating that the corresponding adjective counts have been merged. By contrast, the upper tree in Figure 3.8, which corresponds to the lasso, shows that the solution is sparser

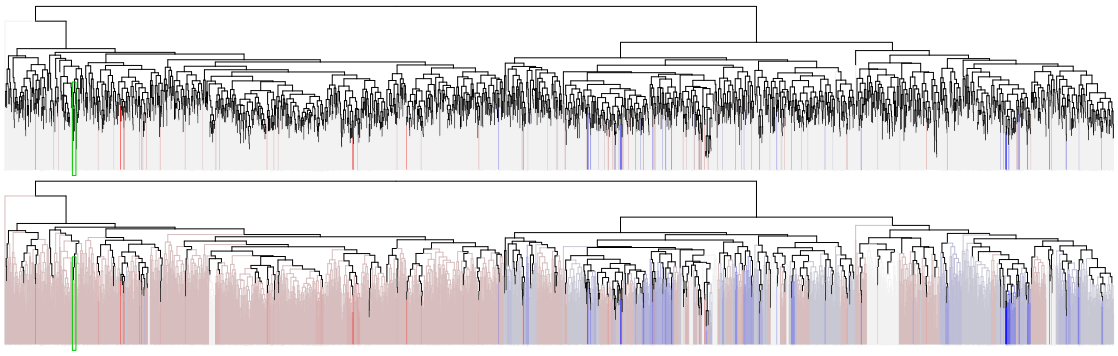

Figure 3.8: Trees for 2,397 adjectives on the leaves with branches colored based on $\hat{\boldsymbol{\beta}}$ estimated with the lasso (Top) and our method (Bottom), respectively. Red branch, blue branch and gray branch correspond to negative, positive and zero $\hat{\beta}_j$, respectively. Darker color indicates larger magnitude of $\hat{\beta}_j$ and lighter color indicates smaller magnitude of $\hat{\beta}_j$.

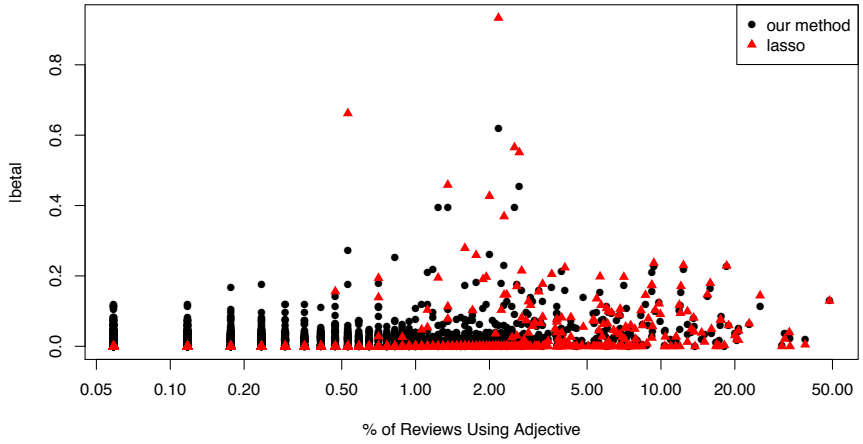


Figure 3.9: $\{|\hat{\beta}_j|\}$ versus term density (on log scale) for adjectives selected by our method (black circles) and the lasso (red triangles) in the $n = 1,700$ and $p = 2,397$ case.

and does not have branches of similar color. Inspection of the merged branches from our method reveals words of similar meaning and sentiment being aggregated. To demonstrate that our method selects rare words whereas the lasso does not, we plot $\{|\hat{\beta}_j|\}$ against the percentage of reviews containing an adjective in Figure 3.9. The rarest word selected by the lasso is "filthy", which appears in 0.47% of reviews. By contrast, our method selects many words that are far more rare: at the extreme of rarest words, our method selects 797 words that appear

in only 0.059% of reviews. Our method is able to select rare words through aggregation. It aggregates 2,244 words into 224 clusters, leaving the remaining 153 words as singletons. Over 70% of these singletons are dense words (where, for this discussion, we call a word "dense" if it appears in at least 1% of reviews and "rare" otherwise). This is four times higher than the percentage of dense words in the original training data. Of the 224 aggregated clusters, 42% are made up entirely of rare words. After aggregation, over half of the clusters become dense features.

Table 3.2 shows the density and estimated coefficient values for eight words falling in a particular subtree of $\mathcal{T}$. The words "heard" and "loud" occur far more commonly than the other six words. We see that the lasso only selects these two words whereas selects all eight words (assigning them negative coefficient values). Our experience with staying at hotels suggests that a review that uses any of the remaining six words would generally be negative. This suggests to us that the lasso's exclusion of these words has to do with their rareness in the dataset rather than their irrelevance to predicting the hotel rating. Indeed, our method selects all eight words and aggregates them into two coefficient values: {heard, loud} with coefficient -0.128 and {yelled, shouted, screaming, crying, blaring, banging} with coefficient -0.039.

Table 3.2: Term density and estimated coefficient for adjectives in the selected group

| adjectives | heard | loud | yelled | shouted | screaming | crying | blaring | banging |
|---|---|---|---|---|---|---|---|---|
| density [4] | 0.0300 | 0.0235 | 0.0006 | 0.0006 | 0.0029 | 0.0006 | 0.0006 | 0.0041 |
| $\hat{\boldsymbol{\beta}}_{\lambda}^{lasso}$ | -0.057 | -0.147 | 0 | 0 | 0 | 0 | 0 | 0 |
| $\hat{\boldsymbol{\beta}}_{\lambda}^{ours}$ | -0.128 | -0.128 | -0.039 | -0.039 | -0.039 | -0.039 | -0.039 | -0.039 |

[4]The term density is computed over train set.

## 3.7 Conclusion

In this chapter, we focus on the challenge posed by highly sparse data matrices, which have become increasingly common in many areas, including biology and text mining. While much work has focused on addressing the challenges of high dimensional data, relatively little attention has been given to the challenges of sparsity in the data. We show, both theoretically and empirically, that not explicitly accounting for the sparsity in the data hurts one's prediction errors and one's ability to perform feature selection. Our proposed method is able to make productive use of highly sparse features by creating new aggregated features based on side information about the original features. In contrast to simpler tree-based aggregation strategies that are occasionally used as a pre-processing step in biological applications, our method adaptively learns the feature aggregation in a supervised manner. In doing so, our methodology not only overcomes the challenges of data sparsity but also produces features that may be of greater relevance to the particular prediction task of interest.

CHAPTER 4

# MICROBIOME COMPOSITIONAL FEATURE SELECTION WITH PHYLOGENETIC TREE

## 4.1 Introduction

The communities of all microbes, including bacteria, fungi and viruses, that inhabit in and on human bodies make up the human microbiome. It is estimated that the cells in human microbiome outnumber human cells by about ten to one (Ley et al., 2006). The gut microbiome community, the largest one among all parts of human body, consists of 100 trillion bacteria in gastrointestinal tract (Devaraj et al., 2013). In addition to the vast number, the gut microbiome plays an important role in human health and disease. Numerous studies reveal contributions of the gut microbiome to metabolism disorders such as obesity, metabolic syndrome and type 2 diabetes (Turnbaugh et al., 2009; Qin et al., 2012; Devaraj et al., 2013). Noto and Peek (2017) show the role of *Helicobacter pylori* bacteria and its interaction with the gut microbiome in the development of gastric cancer. The advancement of next generation sequencing technology contributes to the active research in human microbiome. With the new technology, microbiome composition can be determined by directly sequencing DNA instead of culturing bacteria in lab. There are two popular sequencing techniques: the 16S amplicon sequencing focusing on the 16S rRNA marker gene, and shotgun metagenomic sequencing for all microbial genomic DNA. The first approach is widely used for getting bacterial composition, because 16S rRNA gene is universally present across bacteria and is highly conserved (Li, 2015; Nguyen et al., 2016). The second approach, due to the need of sequencing all

genomic DNA, requires more computation and is found less sensitive in detecting rare taxa (Li, 2015; Tessler et al., 2017). With 16S sequencing, the DNA strands from a variable region of the 16S rRNA gene are counted and mapped to different bacteria, so that the types and abundances of bacteria are determined. The 16S rRNA sequences are clustered into bins called operational taxonomic units (OTUs) based upon similarity (a commonly-used similarity threshold being 97%). One sequence from each OTU cluster is selected as a representative sequence, with which taxonomic ranks are assigned to the entire cluster using a 16S classification method (Wang et al., 2007; Chaudhary et al., 2015). By the end of the pipeline, we get OTU abundances and a phylogenetic tree that encodes taxonomic information of the microbes.

Typically, many OTUs are in extremely low abundances. Researchers often aggregate OTUs to genus or higher taxonomic level to get denser features in preprocessing the data (e.g., Zhang et al. (2012); Chen et al. (2013); Xia et al. (2013); Lin et al. (2014); Randolph et al. (2015); Shi et al. (2016)). Ridenhour et al. (2017) acknowledge the difficulty of estimating highly sparse OTUs due to the lack of variation across samples. However, a large proportion of these aggregated OTUs are eventually discarded because they are found to be still too sparse (e.g., Zhang et al. (2012); Chen et al. (2013); Shi et al. (2016); Wang and Zhao (2017b)). Such aggregation based on arbitrarily decided taxonomic level is clearly *ad hoc*: the implicit assumption that microbiome's functionality stops differentiating beyond genus level is hard to justify in practice. Furthermore, the practice of discarding rare aggregated OTUs potentially wastes useful information. Yan and Bien (2018) argue that a rare feature is not equivalent as an unimportant one since it can be highly predictive to the response. For example, a DNA strand that is associated with a rare disease may have very low

occurrence in general public; however, its occurrence can be highly predictive to the disease. Simply filtering out such rare but predictive microbiome feature is wasteful.

In addition the challenge posed by data rarity, microbiome data is difficult to be modeled statistically due to its compositional nature. Because of the great variability of sequencing reads from sample to sample, microbial abundances are often normalized to relative abundances across taxa. The normalization results in compositional data which preserve a unit sum among all taxa. Such data are not only observed in microbiome analysis: in preprocessing text data, term frequencies are often adjusted for document size, as is noted by Sankaran and Holmes (2017) for the connection between microbiome data and text data. Modeling compositional data is challenging for preserving the unit-sum constraint and accounting for the dependence among the features. BaconShone and Aitchison (1984) propose taking logarithmic transformation on $p$ compositional features and excluding one composition to ensure identifiability. Lin et al. (2014) introduce a variation of the log-contrast model that imposes a zero-sum constraint on regression coefficients in substitute for dropping a feature. With the extra zero-sum constraint, the log-contrast model is equivalent to a $(p-1)$-dimensional model and is thus invariant to the scaling of the counts, making it satisfy the subcomposition coherence principle for compositional data (Aitchison, 1982).

To address the inadequacy of aggregation at pre-determined taxonomic level, we consider a framework that integrates the phylogenetic tree in regression and uses the tree to guide aggregation. Yan and Bien (2018) propose a tree-guided aggregation framework for counts data, where the tree relates features

based on their similarities. In their proposal, the tree as side information can be acquired in various means based on application; in microbiome analysis, a natural tree choice is the phylogenetic tree which encodes evolutionary relationships among the OTUs. The data-driven procedure for aggregating OTUs improves the flexibility in aggregations by allowing them to occur at different taxonomic levels based on their associations with the response. In addition, rare OTUs that would otherwise be discarded in preprocessing can be effectively used by leveraging the phylogenetic tree. In Section 4.2, we tailor the tree-guided aggregation framework to meet the compositional nature of microbiome data. Our adapted framework enables simultaneously fitting a regression and aggregating compositional features. Under the log-contrast model, our framework induces geometric averaging of counts for the OTUs from the same aggregation, whereas the original proposal in Yan and Bien (2018) induces arithmetic averaging through aggregation.

Phylogenetic tree is also widely used to incorporate biological information in other applications. In a linear mixed model setting, Zhai et al. (2018) treat bacterial taxa at different levels as multiple random effects, and compute a kernel matrix for each taxon based on distance measures in the phylogenetic tree. Using the resulting kernel matrices as variance components, they achieve more accurate variable selection performance. In evolutionary biology, Khabbazian et al. (2016) present improved model performance using phylogenetic-tree-based lasso (Tibshirani, 1996) for detecting evolutionary shifts in trait evolution. In particular, they develop a different tree-based parametrization for ours, that can also be used to induce equal coefficients through fusion of subtrees. Both examples illustrate the gain from appropriate use of the phylogenetic tree in modeling.

In this chapter, we use microbiome data from the American Gut project to illustrate the improvement in prediction performance from incorporating phylogenetic tree in microbiome analysis. The data set, processed from fecal samples, includes microbiome composition for 8,120 OTUs across over 1,358 subjects. It also comes with environmental measures such as demographic information, diet information and health information. Our goal is to explain BMI with both gut microbiome compositional features and environmental features. The extreme right-skewness of the distribution of OTU abundances in the left panel of Figure 4.5 indicates that majority OTUs are highly sparse. We compare our tree-guided aggregation framework with $\ell_1$-penalized estimations in (4.2) for the log-contrast model, and show our method achieves better prediction accuracy via effective use of rare OTUs.

We introduce our tree-guided aggregation framework under the log-contrast model in Section 4.2. For our method, we present an efficient alternating direction method of multipliers (ADMM, Boyd et al. (2011)) algorithm that solves our convex optimization problem. In Section 4.3, we use simulations to show the advantage of our method when the underlying true aggregations expand across several taxonomic levels. We also demonstrate the importance of tree completeness by showing the degradation of our method with increasingly distorted phylogenetic tree. In Section 4.4, we present the results from analyzing the gut microbiome data, in which we associate microbiome and environmental features with BMI.

## 4.2 Model and Method

### 4.2.1 The Log-Contrast Model

Microbiome counts data are often normalized to relative abundances at various taxonomic levels. The compositional form preserves a unit sum among all taxa, a constraint that needs to be accounted for in modeling. BaconShone and Aitchison (1984) propose using log-transformed relative abundances as covariates in a linear model and omitting one taxon to ensure identifiability. Lin et al. (2014) introduce a variation of the model that includes all taxa with an additional zero-sum constraint on regression coefficients:

$$y_i = \sum_{j=1}^{p^{\texttt{tax}}} \beta_j^* \log\left(\frac{X_{ij}}{\sum_{j'} X_{ij'}}\right) + \epsilon_i \quad \text{s.t.} \sum_{j=1}^{p^{\texttt{tax}}} \beta_j^* = 0, \tag{4.1}$$

where $\boldsymbol{X}_i = (X_{i1}, \ldots, X_{ip^{\texttt{tax}}})$ are the counts for $p^{\texttt{tax}}$ taxa in the $i$th sample and $\epsilon$ are independent Gaussian errors with mean zero and variance $\sigma^2$. The taxonomic level `tax`, to which people aggregate OTU counts, is determined in preprocessing. Model (4.1) treats all taxa symmetrically by keeping all taxa in the model. Additionally, the model is subcomposition coherent because it is invariant to the scalings of the form $\boldsymbol{X} \to \boldsymbol{D}\boldsymbol{X}$ where $\boldsymbol{D}$ is a diagonal matrix. Subcomposition coherence desires unaltered analysis result when microbiome composition is computed over a subcomposition of the entire taxa (Aitchison, 1982), which are essentially row-wise scalings of $\boldsymbol{X}$. For simplicity, we use $Z_{ij} = X_{ij}/\sum_{j'} X_{ij'}$ to denote the proportion of the $j$th taxon in the $i$th sample in the rest of the chapter.

For typical microbiome data, a large proportion of OTUs are extremely rare and their frequency measures are highly sparse. In order to get denser data, researchers often aggregate OTUs to genus or higher taxonomic level and dis-

card sparse aggregated OTUs. After these preprocessing steps, Lin et al. (2014) propose an $\ell_1$-penalized estimation for Model (4.1):

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^{p^{\texttt{tax}}}} \left\{ \frac{1}{2n} \sum_{i=1}^{n} \left( y_i - \sum_{j=1}^{p^{\texttt{tax}}} \log\left(Z_{ij}\right)\beta_j \right)^2 + \lambda \|\boldsymbol{\beta}\|_1 \;\; \text{s.t.} \sum_{j=1}^{p^{\texttt{tax}}} \beta_j = 0 \right\}. \tag{4.2}$$

The rationale for the aggregation and elimination of rare microbiome data is to ensure enough variability across samples so that the features can be well estimated. However, the strategy of determining a single taxonomic level for aggregation lacks a theoretical support and is not data-driven. In particular, the success of such strategy relies on an implicit assumption that microbiome's functionality stops differentiating beyond the chosen `tax` level. Moreover, the practice of discarding rare aggregated OTUs can be wasteful, because a rare feature is not equivalent as an unimportant one (Yan and Bien, 2018). Our goal is to introduce a data-driven aggregation framework that avoids setting hard threshold for discarding rare features. Next, we propose integrating the phylogenetic tree into the estimation problem to guide the aggregations.

### 4.2.2 Phylogenetic-Tree-Guided Aggregation

A phylogenetic tree grows upon OTUs and expands across several taxonomic levels: species, genus, family, order, class, phylum and kingdom. It relates OTUs based their similarities and encodes their evolutionary relationships. With tree $\mathcal{T}$ and node $u$, let $\mathcal{T}_u$ be the subtree rooted by $u$ and let $\text{Ł}(\mathcal{T}_u)$ be the leaf set of $\mathcal{T}_u$. In a linear model setting, Yan and Bien (2018) note the equivalence between aggregating features on the leaves and enforcing equal $\beta$ values in the

subtree. With `tax` being the OTU level, the equivalence under Model (4.1) is

$$\text{aggregating} \sum_{j \in Ł(\mathcal{T}_u)} \beta_j \log(X_{ij}) = \tilde{\beta} \sum_{j \in Ł(\mathcal{T}_u)} \log(X_{ij}) \quad \Leftrightarrow \quad \text{enforcing } \beta_j = \tilde{\beta}\ \forall j \in Ł(\mathcal{T}_u).$$

In order to fuse $\beta_j$'s within a subtree, Yan and Bien (2018) develop a tree-based parametrization that assigns a latent variable $\gamma_u$ to every node $u$ of the tree and associates $\beta_j$ with its ancestor $\gamma_u$'s through

$$\beta_j = \sum_{u \in \text{ancestor}(j) \cup \{j\}} \gamma_u, \tag{4.3}$$

where ancestor$(j) \cup \{j\}$ is the set of nodes on the path from the root of the tree to the leaf of $\beta_j$. The relationship can be expressed more concisely as $\boldsymbol{\beta} = \boldsymbol{A}\boldsymbol{\gamma}$, where $\boldsymbol{A} \in \{0,1\}^{p \times |\mathcal{T}|}$ is a binary matrix encoding ancestor-descendant relationships between leaves (in the rows of $\boldsymbol{A}$) and all tree nodes (in the columns of $\boldsymbol{A}$): $A_{jk} = 1_{\{u_k \in \text{ancestor}(j) \cup \{j\}\}}$. The top panel of Figure 4.1 illustrates the tree-based parametrization in a phylogenetic tree. The colored $\beta_j$'s on the leaves indicate group membership when aggregations occur at genus level, as is commonly done in practice. This tree-based parametrization gives us more flexibility by allowing aggregations occur at multiple taxonomic levels of the tree. In the bottom panel of Figure 4.1, zeroing out $\gamma_u$'s in the crossed nodes leads to aggregation at the shaded levels.

A key difference between the original proposal in Yan and Bien (2018) and our aggregation under log-contrast model is how count features are averaged. Suppose we let $\beta_j = \tilde{\beta}\ \forall j \in Ł(\mathcal{T}_u)$. In Yan and Bien (2018), aggregation is equivalent to taking arithmetic mean for all OTU counts in subtree $\mathcal{T}_u$, i.e., $\sum_{j \in Ł(\mathcal{T}_u)} \beta_j X_{ij} = \tilde{\beta} \cdot |Ł(\mathcal{T}_u)| \cdot (\sum_{j \in Ł(\mathcal{T}_u)} X_{ij} / |Ł(\mathcal{T}_u)|)$. Since we model compositional data using a log-contrast model, our features are of log-transformed form. In

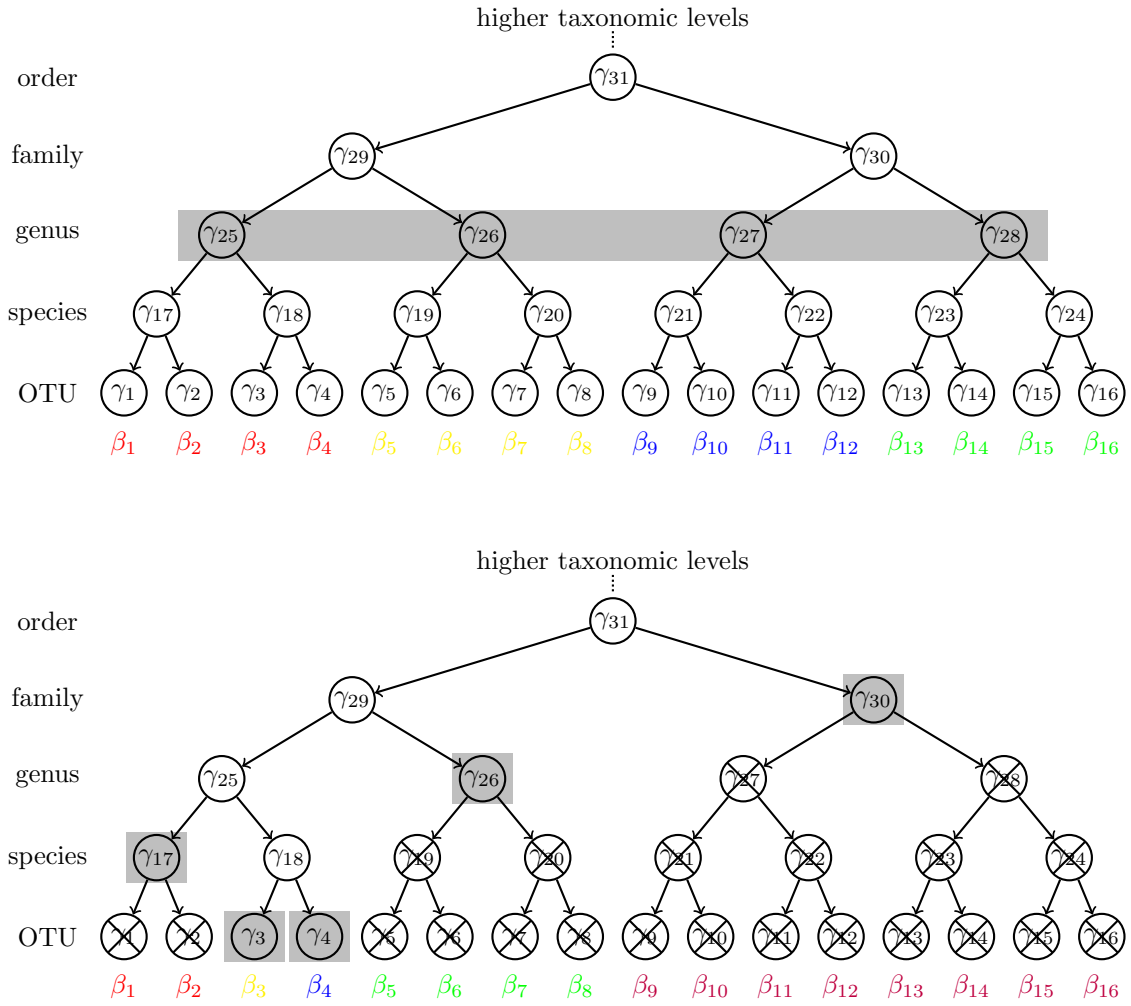


Figure 4.1: (Top) phylogenetic tree with $\gamma_u$ assigned to node $u$ and a taxonomic level labeled for every depth. When aggregating OTU counts to genus level (shaded in gray), the OTUs naturally separate into four subtrees with leaves colored accordingly. (Bottom) A more flexible aggregation pattern induced by zeroing out $\gamma_u$'s in the crossed nodes. The roots of aggregated subtrees are shaded in gray and the corresponding OTUs are colored accordingly on the leaves. In both examples, $\beta_j$'s from the same aggregation share equal values.

our case, aggregation is the same as taking geometric mean for all OTUs in $\mathcal{T}_u$:

$$\sum_{j\in Ł(\mathcal{T}_u)} \beta_j \log(X_{ij}) = \tilde{\beta} \sum_{j\in Ł(\mathcal{T}_u)} \log(X_{ij}) = \tilde{\beta} \cdot |Ł(\mathcal{T}_u)| \cdot \log\left(\left(\prod_{j\in Ł(\mathcal{T}_u)} X_{ij}\right)^{1/|Ł(\mathcal{T}_u)|}\right).$$

Our proposed estimator is a solution to the following optimization problem:

$$\min_{\substack{\boldsymbol{\beta}\in\mathbb{R}^p \\ \text{and } \boldsymbol{\gamma}\in\mathbb{R}^{|\mathcal{T}|}}} \left\{ \frac{1}{2n} \left\| \boldsymbol{y} - \log(\boldsymbol{X})\boldsymbol{\beta} \right\|_2^2 + \lambda_1 \|\boldsymbol{\gamma}_{-\text{root}}\|_1 + \lambda_2 \|\boldsymbol{\beta}\|_1 \text{ s.t. } \boldsymbol{\beta} = \boldsymbol{A}\boldsymbol{\gamma} \text{ and } \mathbf{1}^T\boldsymbol{\beta} = 0 \right\}. \tag{4.4}$$

Since the log-contrast model is invariant to scalings of $\boldsymbol{X}$, working with counts features $\boldsymbol{X}$ or relative abundances $\boldsymbol{Z}$ will yield the same solution for (4.4). In preprocessing, we impute $\boldsymbol{X}$ by replacing zero counts with 0.5 before taking logarithmic transformation on $\boldsymbol{X}$ coordinates. de la Cruz and Kreft (2018) acknowledge the difficulty of using the geometric mean when there are zeros, and propose choosing a pseudocount based on data itself. In Problem (4.4), the relative size of $\lambda_1$ and $\lambda_2$ controls the tradeoff between fusion and sparsity of $\boldsymbol{\beta}$: larger $\lambda_1$ induces sparsity in $\boldsymbol{\gamma}$ and thus increases the fusion level of $\boldsymbol{\beta}$. In practice, we learn optimal $(\lambda_1, \lambda_2)$ using cross validation.

Our method's success depends on the quality of the phylogenetic tree. With 16S sequencing, the taxonomic lineage to an OTU can be assigned by comparing its characteristic sequence to a known 16S rRNA database. However, many OTUs will not have their sequence matched in the database. For example, in the gut microbiome data that we study in Section 4.4, 89%, 49% and 13% of the OTUs do not have species label, genus label and family label, respectively. The missing taxonomic information for many OTUs translates into a low-quality phylogenetic tree, which may hinder our method's performance. In Section 4.3, we explore the degradation of our method with the amount of tree distortion from missing taxonomic labels.

In terms of computation, we follow Yan and Bien (2018) to develop an alternating direction method of multipliers (ADMM, Boyd et al. (2011)) to efficiently solve (4.4). To start with, we write (4.4) as a global consensus problem that decouples different parts of the probelm by introducing copies of $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$:

$$\min_{\substack{\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{\beta}^{(3)},\boldsymbol{\beta}\in\mathbb{R}^p \\ \text{and } \boldsymbol{\gamma}^{(1)},\boldsymbol{\gamma}^{(2)},\boldsymbol{\gamma}\in\mathbb{R}^{|\mathcal{T}|}}} \left\{ \frac{1}{2n} \left\| \boldsymbol{y} - \log(\boldsymbol{X})\boldsymbol{\beta}^{(1)} \right\|_2^2 + \lambda_1 \|\boldsymbol{\gamma}^{(1)}_{-\text{root}}\|_1 + \lambda_2 \|\boldsymbol{\beta}^{(2)}\|_1 \right\}$$

$$\text{s.t.} \begin{pmatrix} \boldsymbol{I}_p \\ \boldsymbol{1}^T \end{pmatrix} \boldsymbol{\beta}^{(3)} = \begin{pmatrix} \boldsymbol{A} \\ \boldsymbol{0}^T \end{pmatrix} \boldsymbol{\gamma}^{(2)}, \boldsymbol{\beta} = \boldsymbol{\beta}^{(1)} = \boldsymbol{\beta}^{(2)} = \boldsymbol{\beta}^{(3)} \text{ and } \boldsymbol{\gamma} = \boldsymbol{\gamma}^{(1)} = \boldsymbol{\gamma}^{(2)}. \tag{4.5}$$

We let $\rho$ be the parameter for quadratic penalties in the Lagrangian of (4.5) (see (C.1) of Appendix C.0.1). Conventional ADMM uses fixed penalty parameter throughout its iterations; however, many studies have reported the sensitivity issue of ADMM to the penalty parameter. Xu et al. (2017) propose an adaptive updating scheme for the penalty parameter based upon local sharpness property of an objective function. The locally-adaptive ADMM (LA-ADMM) updates $\rho$ after $N_{inner}$ iterations of conventional ADMM, and Xu et al. (2017) proves the convergence of LA-ADMM after $N_{outer}$ updates of $\rho$. We notice from our experience in numerical studies that $N_{inner} = 1,000$ and $N_{outer} = 30$ yield promising convergence for LA-ADMM. See Algorithm 9 and Algorithm 5 in Appendix C.0.1 for details of LA-ADMM. In particular, Algorithm 5 contains conventional ADMM updates that are derived in Appendix C.0.1. These updates involve performing matrix multiplies and soft-thresholdings. The computationally-involved parts, singular value decomposition (SVD) of $\log(\boldsymbol{X})$ and SVD of $\begin{pmatrix} \boldsymbol{I}_p & -\boldsymbol{A} \\ \boldsymbol{1}^T & \boldsymbol{0}^T \end{pmatrix}$, need only be computed once and can be reused for all $\rho$, $\lambda_1$ and $\lambda_2$.

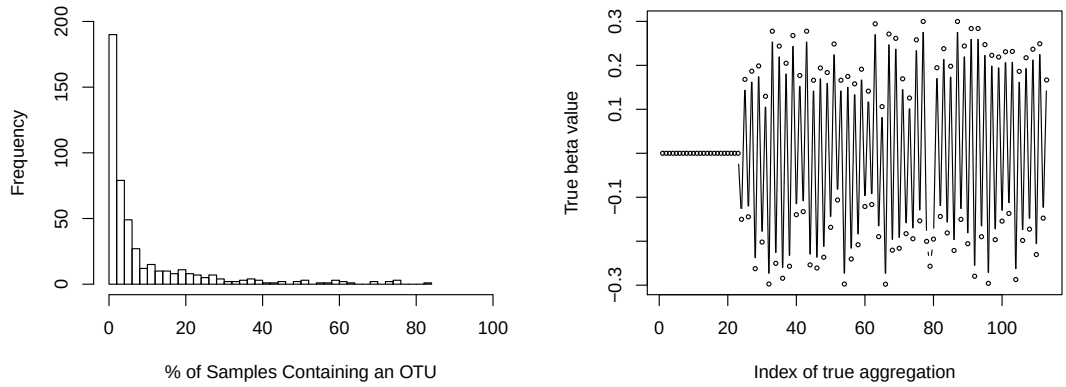


Figure 4.2: (Left) Distribution of OTU densities for 481 OTUs in 100 samples used in simulation. (Right) Generated $\tilde{\boldsymbol{\beta}}^*$ elements for the 113 true aggregations.

## 4.3 Simulation Study

In Section 4.2, we show statistically how our aggregation framework is more flexible than traditional practice of aggregation at a single pre-determined taxonomic level. To compare these methods numerically, we construct a scenario in which *true aggregations* of log-transformed OTUs are well-defined, and the aggregation levels range from OTU to family. In order to mimic real-world situation, we subset 481 OTUs with complete phylogenetic lineage and 100 samples (i.e., $n = 100$ and $p = 481$) from the gut microbiome data in Section 4.4. The left panel of Figure 4.2 shows that the OTUs are relatively sparse: more than 50% of them appear in less than 5% of the samples. We construct a phylogenetic tree for the OTUs from the accompanying taxonomy matrix and visualize the tree in Figure 4.3. We form $\boldsymbol{A} \in \{0,1\}^{481\times 858}$ for the tree which has 481 leaves and 858 nodes (including leaves).

We specify the true aggregations by (roughly) equally splitting OTUs for being aggregated at species, genus, family, order or class level, or being left unaggregated. We end up with 113 aggregations (including OTU singletons)

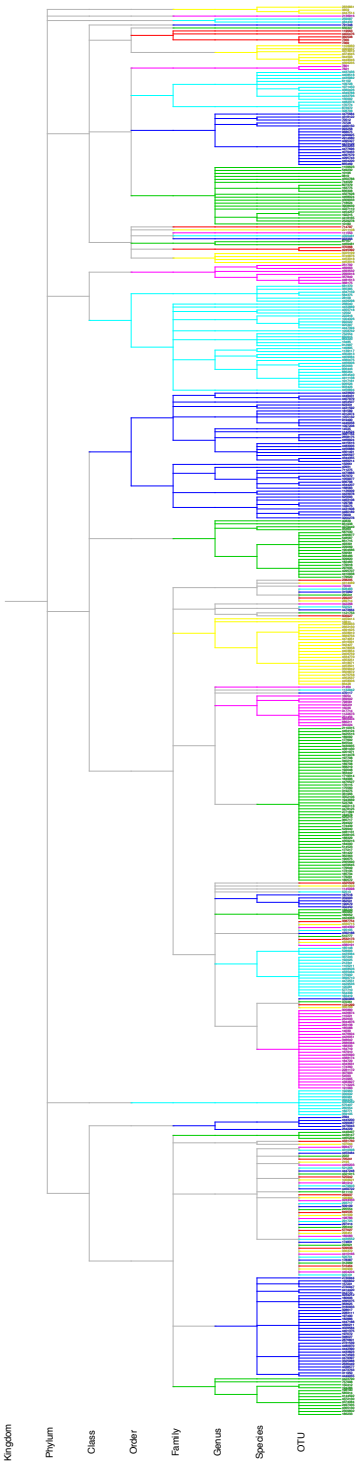


Figure 4.3: Phylogenetic tree built upon taxonomy matrix for the 481 OTUs used in simulation. OTU labels are on the leaves. There are seven taxonomic levels upon OTU: species, genus, family, order, class, phylum and kingdom; each level corresponds to a depth in the tree. The 481 OTUs are either kept at OTU level by themselves or aggregated up to the class level. Subtrees are colored accordingly to illustrate the true aggregations.

and their corresponding subtrees are colored accordingly in Figure 4.3. Let $B^*$ be the set of roots of these subtrees. Let $\boldsymbol{A}_{B^*} \in \{0,1\}^{481\times 113}$ be the column subset of $\boldsymbol{A}$ for the columns corresponding to the roots in $B^*$. By definition of $\boldsymbol{A}$, OTU $j$ is in the aggregation specified by subtree $\mathcal{T}_u$ if and only if $[A_{B^*}]_{ju} = 1$. We generate $\boldsymbol{\beta}^* = \boldsymbol{A}_{B^*}\tilde{\boldsymbol{\beta}}^*$ where the first 20% elements of $\tilde{\boldsymbol{\beta}}^* \in \mathbb{R}^{113}$ are zero. The remaining elements of $\tilde{\boldsymbol{\beta}}^*$ are drawn independently from Unif$(-0.1, 0.1)$, with alternating -0.2 and 0.2 being added to neighboring $\tilde{\beta}^*_u$'s. The added alternating values are to make sure sibling aggregations have distinct enough signals to be well estimated. Finally, we rescale the $\tilde{\beta}^*_u$ corresponding the largest aggregation so that $\mathbf{1}^T\boldsymbol{\beta}^* = 0$. The right panel of Figure 4.2 shows the elements of $\tilde{\boldsymbol{\beta}}^*$ versus index of true aggregation. The response $\boldsymbol{y} \in \mathbb{R}^n$ is simulated from (4.1) with $\sigma^2 = \|\log(\boldsymbol{Z})\boldsymbol{\beta}^*\|_2^2/(n \cdot \texttt{snr})$ where `snr` is signal-to-noise ratio. For each method in the following simulations, we average its performance over 10 repetitions.

We apply our method with $\boldsymbol{A}$ and compute it over a 20-by-20 grid of $(\lambda_1, \lambda_2)$ values. As a comparison, we consider an *oracle method* that is tuned over a length-20 grid of $\lambda$ values,

$$\hat{\boldsymbol{\beta}}^{\text{oracle}} = \underset{\{\boldsymbol{\beta}\in\mathbb{R}^p:\boldsymbol{\beta}=\boldsymbol{A}_{B^*}\tilde{\boldsymbol{\beta}} \text{ for } \tilde{\boldsymbol{\beta}}\in\mathbb{R}^{206}\}}{\arg\min} \left\{ \frac{1}{2n}\left\|\boldsymbol{y} - \log(\boldsymbol{Z})\boldsymbol{\beta}\right\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1 \quad \text{s.t. } \mathbf{1}^T\boldsymbol{\beta} = 0 \right\}.$$

The oracle method yields the best attainable result that any method can achieve in estimating $\boldsymbol{\beta}^*$. We also consider *unsupervised aggregations* that aggregate log-transformed OTUs to a pre-determined `tax` level. For `tax` $\in$ {species, genus, family}, we let $\boldsymbol{A}^{\texttt{tax}} \in \{0,1\}^{p\times p^{\texttt{tax}}}$ be the column subset for columns corresponding to tree nodes at the `tax` level. In unsupervised aggregations, we aggregate log-transformed OTUs to the `tax` level and then apply an $\ell_1$-penalized estimation over the aggregated data,

$$\hat{\boldsymbol{\beta}}^{\texttt{tax}} = \boldsymbol{A}^{\texttt{tax}}\tilde{\boldsymbol{\beta}}^{\texttt{tax}} \quad \text{for} \quad \tilde{\boldsymbol{\beta}}^{\texttt{tax}} = \underset{\tilde{\boldsymbol{\beta}}\in\mathbb{R}^{p^{\texttt{tax}}}}{\arg\min} \left\{ \frac{1}{2n}\left\|\boldsymbol{y} - \log(\boldsymbol{Z})\boldsymbol{A}^{\texttt{tax}}\tilde{\boldsymbol{\beta}}\right\|_2^2 + \lambda\|\tilde{\boldsymbol{\beta}}\|_1 \quad \text{s.t. } \mathbf{1}^T\tilde{\boldsymbol{\beta}} = 0 \right\}.$$

We tune the above problem over a length-20 grid of $\lambda$ values. When `tax` is OTU level, the raw compositions are modeled without being aggregated since $\boldsymbol{A}^{\texttt{tax}} = \boldsymbol{I}_p$. To evaluate each method's performance, we use *best mean-squared estimation error*, i.e., $\min_\Lambda \|\hat{\boldsymbol{\beta}}(\Lambda) - \boldsymbol{\beta}^*\|_2^2/p$, and *best mean-squared prediction error*, i.e., $\min_\Lambda \|\hat{\boldsymbol{y}}(\Lambda) - \log(\boldsymbol{Z})\boldsymbol{\beta}^*\|_2^2/n$, where each method's best performance is evaluated on respective tuning parameter set $\Lambda$. The predictions are made accordingly for the methods: our method and the oracle method estimate $\boldsymbol{y}$ with $\log(\boldsymbol{Z})\hat{\boldsymbol{\beta}}$, whereas it is $\log(\boldsymbol{Z})\boldsymbol{A}^{\texttt{tax}}\tilde{\boldsymbol{\beta}}^{\texttt{tax}}$ the unsupervised aggregations.

We first investigate these methods' performance with varying signal-to-noise ratio `snr`. As `snr` $\in \{10, 20, 30, 40, 50\}$ increases, the simulated response has more signal as opposed to noise. The top panels of Figure 4.4 illustrate the performance of all the methods in estimation and in prediction, respectively. Our method, which learns aggregation through $\boldsymbol{A}$, outperforms the fits with unsupervised aggregations at all `tax` levels. As `snr` increases, all the methods yield better estimation and prediction. In the top right panel, the higher taxonomic level aggregations occur, the better prediction performance is achieved for unsupervised aggregations. Given that a third of the true aggregations occur above family level, family-level aggregation, which is a typical choice in many microbiome studies, is closer to the truth than aggregations at even lower taxonomic levels. The fit with family-level aggregations even slightly outperforms our method in prediction. As `snr` becomes large enough, our method and the fit with family-level aggregations converge to the oracle.

So far we evaluate our method's performance in the ideal situation that a complete phylogenetic tree is available and all the OTUs have complete taxonomic labels. However, due to the limitation of sequence classification methods,

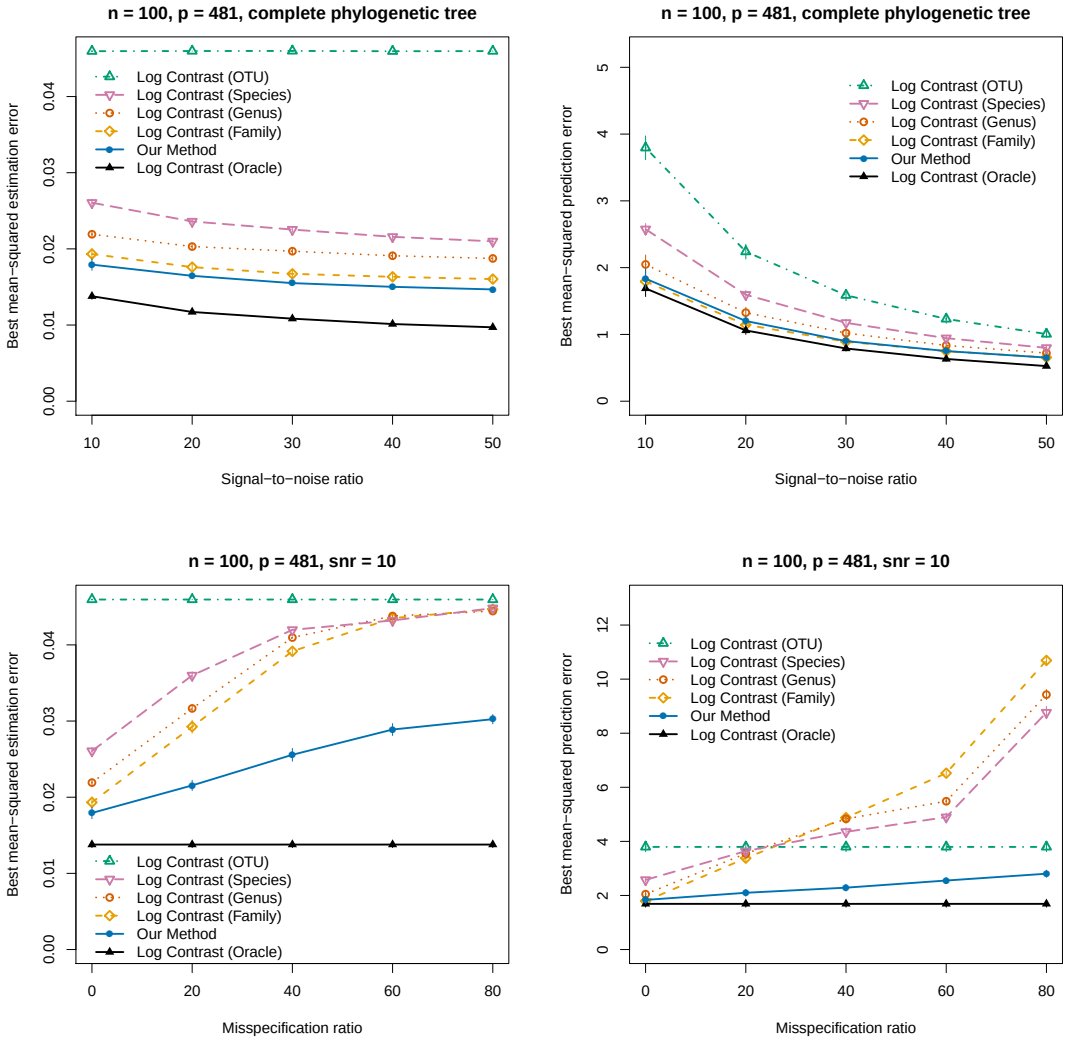


Figure 4.4: (Top Left) $\min_\Lambda \|\hat{\boldsymbol{\beta}}(\Lambda)-\boldsymbol{\beta}^*\|_2^2/p$ and (Top Right) $\min_\Lambda \|\hat{\boldsymbol{y}}(\Lambda)-\log(\boldsymbol{Z})\boldsymbol{\beta}^*\|_2^2/n$ versus varying `snr`. (Bottom) Fixing `snr` = 10, (Bottom Left) $\min_\Lambda \|\hat{\boldsymbol{\beta}}(\Lambda)-\boldsymbol{\beta}^*\|_2^2/p$ and (Bottom Right) $\min_\Lambda \|\hat{\boldsymbol{y}}(\Lambda) - \log(\boldsymbol{Z})\boldsymbol{\beta}^*\|_2^2/n$ versus increasing proportions of OTUs missing species, genus and family labels.

it is common for many OTUs to have missing taxonomic labels in microbiome data. For the gut microbiome data that we analyze in Section 4.4, 89% of the OTUs do not have complete taxonomic labels, where the majority of the missing labels occur on or below family level. Thus, the phylogenetic tree that we have access to in practice is only an approximation to the ideal complete tree. We next study the sensitivity of our method to the completeness of the phyloge-

netic tree. We generate a sequence of distorted trees by removing species, genus and family labels for increasing proportions of OTUs. As the proportion increases from zero to 80%, we nest the OTUs that are previously selected within the larger set. The resulting tree becomes less informative for larger proportion. The bottom panels of Figure 4.4 show the degradation of our method as more parts of the tree become missing. When 80% of the OTUs have missing taxonomic labels up to family level, the remaining tree only allows aggregations to order level for 80% of the OTUs. As the tree gets more distorted, all the methods that rely on aggregations yield worse performance in estimation and prediction. Our method degrades much more slowly than the other methods, indicating its resistance to misspecified trees.

## 4.4 Gut Microbiome Analysis

Numerous studies recognize gut microbiome for its important role in several human metabolism diseases (e.g., Turnbaugh et al. (2009); Qin et al. (2012); Devaraj et al. (2013)). Unveiling the function mechanism and interactions of a vast number of microbes in human gut is beneficial to public health. In this study, we aim to model BMI with gut microbiome compositions and several environmental covariates, in which we show the merit of our method.

The American Gut is one of the largest crowd sourced projects in the United States working on bridging human microbiome and health. In a data set from the American Gut, we have sequencing data for over 27 thousands OTUs across 8 thousands subjects. In addition, the data set comes with environmental covariates for each subject, including demographic information, diet information and

health information. We adopt 8 environmental covariates, including *Age, Sex, Milk and Cheese Frequency, Red Meat Frequency, Sugar-Sweetened Drink Frequency, Fruit Frequency, Vegetable Frequency* and *Whole Grain Frequency*, in modeling the mean of BMI under (4.1): for subject $i$,

$$\mathbb{E}[\text{BMI}_i] = \log(\boldsymbol{Z}_i)\boldsymbol{\beta}^* + \boldsymbol{W}_i\boldsymbol{\eta}^* \quad \text{s.t. } \mathbf{1}^T\boldsymbol{\beta}^* = 0,$$

where $\boldsymbol{Z}_i \in \mathbb{R}^{p^{\texttt{tax}}}$ are compositions for $p^{\texttt{tax}}$ taxa at `tax` level, and $\boldsymbol{W}_i \in \mathbb{R}^9$ are the 8 environmental covariates and an intercept. We avoid specifying the value for `tax` because truth may vary for different models: for our method which aggregates from OTU level, `tax` is OTU level; for the traditional approach that works on genus-level taxa, `tax` is genus level.

After filtering out missing values in BMI and the 8 environmental covariates, we end up with 8,120 OTUs and 1,358 samples. The OTUs are very sparse: more than 80% of them appear in less than 5% of the samples. The left panel of Figure 4.5 shows the rarity of microbes at OTU level. In many studies, a commonly adopted[1] preprocessing step involves aggregating microbiome abundances to genus level and discarding highly sparse genera (e.g., Zhang et al. (2012); Chen et al. (2013); Shi et al. (2016); Wang and Zhao (2017b)). We follow the preprocessing step to generate genus-level data for a comparison method to ours. After the preprocessing in which we discard any genera appearing in less than 5% of the samples, there are 980 genera left whose densities are shown in the middle panel of Figure 4.5.

For our method, we construct the phylogenetic tree (and hence $\boldsymbol{A}$) from the accompanied taxonomy matrix for the 8,120 OTUs. Many OTUs have missing taxonomic labels from the taxonomy matrix: 89%, 49% and 13% of the OTUs

[1] yet criticized by us

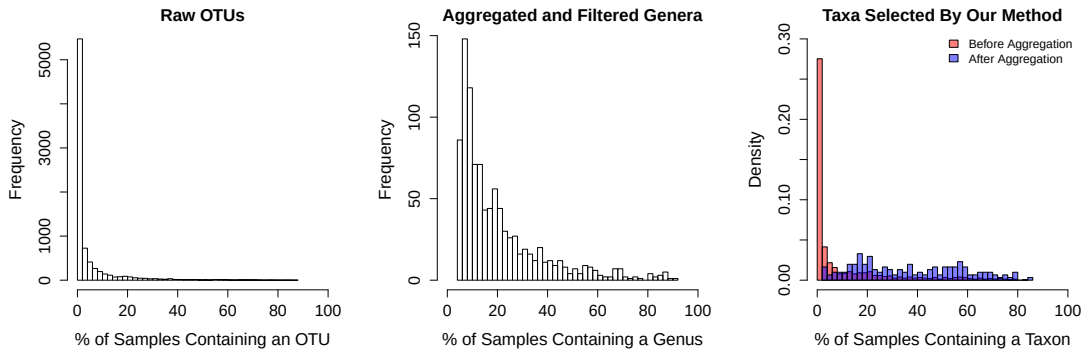


Figure 4.5: In the gut microbiome data, distribution of microbe densities at OTU level (Left) and at genus level (Middle) (after filtering at 5% threshold of density). Our method at its best performance aggregates 761 OTUs into 152 taxa with non-zero coefficients. The right panel overlays the densities of the 761 OTUs (in red) and that of the 152 taxa (in blue).

| **Method** | $p^{\texttt{tax}}$ | **Model Size (CV)** | **Test Error (CV)** | **Model Size (Best Possible)** | **Test Error (Best Possible)** |
|---|---|---|---|---|---|
| Our Method | 8,120 | 51 | **16.06** | 152 | **15.43** |
| (4.2) at OTU Level | 8,120 | 48 | 16.11 | 91 | 15.71 |
| (4.2) at Genus Level | 980 | 48 | 16.13 | 93 | 15.81 |

Table 4.1: Test set MSE and model size for CV-chosen fit and best-performing fit

do not have species label, genus label and family label, respectively. The resulting incomplete phylogenetic tree from the missing labels may hinder our method's performance. We consider two $\ell_1$-penalized methods as a comparison to ours: (4.2) with `tax` being OTU level and (4.2) with `tax` being genus level. We include an intercept and 8 environmental covariates in the squared loss of (4.4) and (4.2), without having their regression coefficients penalized. The adjusted computational paths for including these non-microbiome covariates can be found in Appendix C.0.2. We hold out 20% of the data as test set and train each method on the remaining data using 5-fold cross validation (CV). We vary the tuning parameters $(\lambda_1, \lambda_2)$ over a 10-by-10 grid of values for our method, and tune (4.2) with a length-20 grid of $\lambda$ values.

We evaluate the methods' performance using test set *mean squared error*

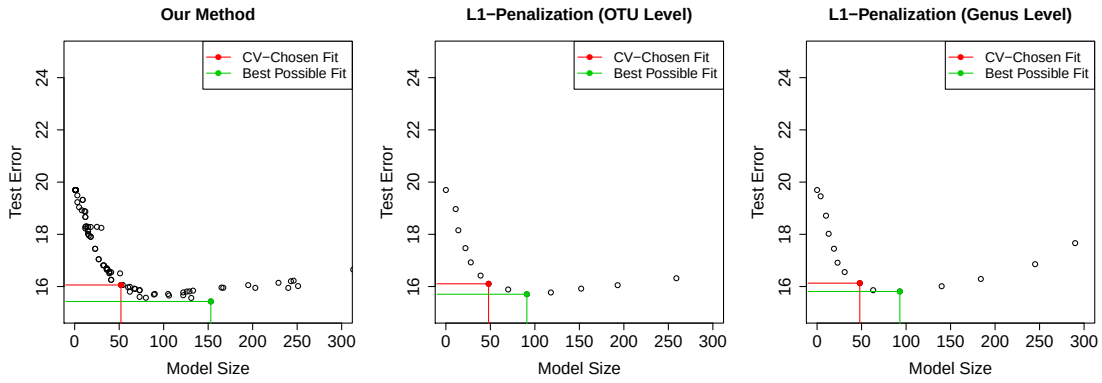


Figure 4.6: (Left) Test set MSE versus model size at every $(\lambda_1, \lambda_2)$ of our method, where model size for (4.4) is the number of resulting aggregations with non-zero coefficients. At OTU level (Middle) and at genus level (Right), Test set MSE versus model size at every $\lambda$ of (4.2), where model size for (4.2) is the number of selected taxa. Red and blue points correspond to the tuning parameters selected by CV and that achieves the lowest test error, respectively.

(MSE) and *model size* in Table 4.1. The model size of (4.2) at a given $\lambda$ is the number of selected microbiome taxa in the corresponding fit. As our method induces both aggregation and sparsity in OTUs, its model size at a given $(\lambda_1, \lambda_2)$ is the number of recovered aggregations with non-zero regression coefficients. From Table 4.1, (4.2) yields better predictions when working at OTU level instead of at genus level, which acts against the current preprocessing decision of aggregating OTU to genus (or higher) level. The improvement of our method from (4.2) at OTU level shows the merit of task-driven aggregation under our framework. The three methods have similar model size at respective CV-chosen fit. However, at the best performing fit, our method recovers substantially more aggregated features than (4.2). Figure 4.6 show the relationship between test error and model size for the three methods. The bell-shaped curves indicate that the methods are well-tuned with the current choice of tuning parameters.

In the following analyses, we focus on the best possible performance for each of the three methods. To demonstrate that our method selects rare OTUs

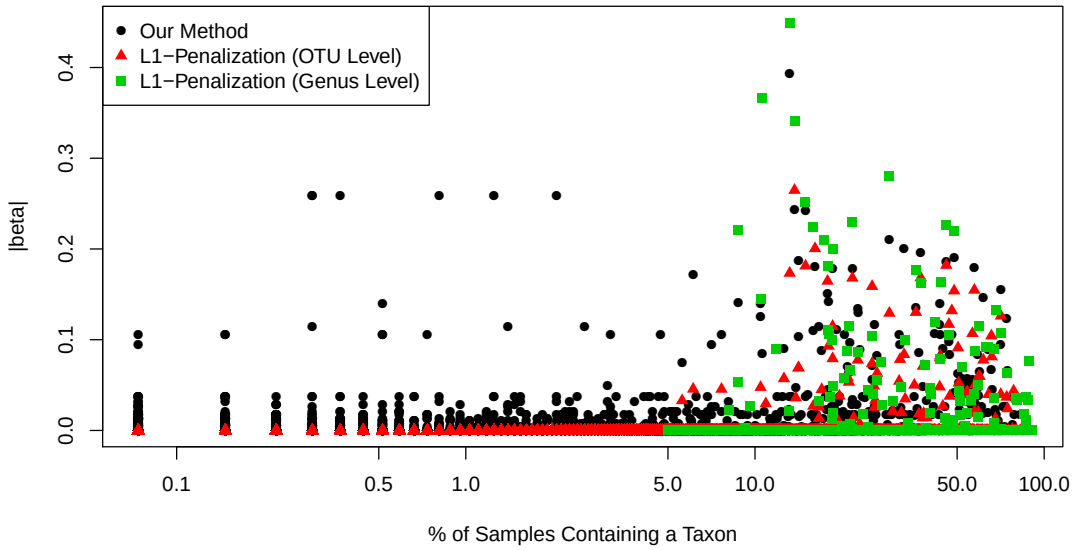


Figure 4.7: $\{|\hat{\beta}_j|\}$ versus density (on log scale) for OTUs selected by our method (black circles) and (4.2) at OTU level (red triangles), for genera selected by (4.2) at genus level (green squares).

whereas lasso does not, we plot $\{\hat{\beta}_j\}$ against the percentage of samples containing a taxon in Figure 4.7. For our method and (4.2) at OTU level, the corresponding taxa are OTUs; for (4.2) at genus level, the corresponding taxa are filtered genera. Our method selects several OTUs appearing in 0.07% of samples, whereas the (4.2) only selects OTUs appearing in more than 5% of samples. Our tree-guided aggregation framework effectively aggregates OTUs to denser data. At its best performance, our method selects 761 OTUs and aggregates them into 152 groups (including singleton OTUs). In the right panel of Figure 4.5, we overlay two density plots for taxa densities: the red one for the 761 selected OTUs before aggregation and the blue one for the 152 aggregated taxa. It is clear that the selected taxa become denser after being aggregated by our framework. Moreover, the high stake in the right panel of Figure 4.5 indicates that these selected taxa were highly sparse before being aggregated.

The tree-guided aggregation framework also leads to more interpretable re-

sult with aggregation decisions made based on OTUs' biological relevance to model task. Of the 761 OTUs selected by our method, 636 of them are aggregated to 27 groups on or above species level; the remaining 125 OTUs are recovered as OTU singletons. We summarize the 27 aggregated groups in Table 4.2, including information on aggregation level, corresponding taxa, size of aggregation and density before and after aggregation. Most of the aggregations occur at species and genus level, with a small amount at family level. The size of aggregations at those levels tend to be below 40, except for some large species (e.g., *Prausnitzii*) and large family (e.g., *Lactobacillaceae*). There is one aggregation at order level (*Fusobacteriales*) and one aggregation at phylum level (*Cyanobacteria*). Among the four genera identified as significant in predicting BMI by Lin et al. (2014) and Shi et al. (2016), we successfully recovered three of them: *Acidaminococcus*, *Clostridium* and *Alistipes* (as part of *Rikenellaceae* family-level aggregation). Most of the aggregations deal with OTUs under a wide range of rarities; 22 out of 27 aggregations contain an OTU that appears in 0.07% of samples. Meanwhile, every aggregation has at least one dense OTU, indicating that the aggregation framework adds rare OTUs to dense ones. The last column compares OTUs selected by our method and those selected by (4.2) at OTU level. The lasso selects 91 OTUs, out of which 86 are part of the 761 OTUs selected by our method. However, only 10 of the shared OTUs come from one of our aggregations (noticing the vast zero entries in the last column).

Many of the selected aggregations in Table 4.2 are known to be associated with BMI; we now look into a specific microbe called *Akkermansia muciniphila*. *Akkermansia muciniphila* is found to have decreased abundance in obese and diabetic patients in several studies (Karlsson et al., 2012; Santacruz et al., 2010; Zhang et al., 2013), and its effect in glucose metabolism is well-understood

| Aggreg. Level | Aggregated Taxon | Size of Aggreg. | $\hat{\beta}^{\text{ours}}$ | OTU Density L.B. (%) | OTU Density U.B. (%) | Aggreg. Density (%) | No. Shared OTUs (lasso) |
|---|---|---|---|---|---|---|---|
| Species | Muciniphila | 11 | -0.1059 | 0.07 | 43.74 | 68.11 | 2 |
| Species | Producta | 19 | -0.0377 | 0.07 | 13.18 | 28.13 | 0 |
| Species | Copri | 34 | -0.0208 | 0.07 | 25.77 | 52.43 | 1 |
| Species | Prausnitzii | 71 | -0.0180 | 0.07 | 82.62 | 85.42 | 1 |
| Species | Parvula | 5 | -0.0004 | 0.29 | 9.94 | 11.19 | 0 |
| Species | Ovatus | 7 | 0.0153 | 0.07 | 38.29 | 47.05 | 1 |
| Species | Uniformis | 16 | 0.0263 | 0.07 | 73.34 | 79.46 | 0 |
| Species | Bromii | 3 | 0.0360 | 0.15 | 57.44 | 57.44 | 1 |
| Species | Eggerthii | 3 | 0.0950 | 0.07 | 31.74 | 31.81 | 1 |
| Genus | Bifidobacterium | 22 | -0.0369 | 0.07 | 35.57 | 55.82 | 1 |
| Genus | Dysgonomonas | 11 | -0.0322 | 0.07 | 19.44 | 20.77 | 0 |
| Genus | Anaerostipes | 13 | -0.0291 | 0.07 | 22.53 | 41.46 | 0 |
| Genus | Rothia | 21 | -0.0173 | 0.07 | 26.22 | 34.61 | 0 |
| Genus | Serratia | 15 | -0.0024 | 0.07 | 28.13 | 33.21 | 0 |
| Genus | Collinsella | 9 | 0.0006 | 0.07 | 54.71 | 57.88 | 0 |
| Genus | Bilophila | 4 | 0.0010 | 0.07 | 64.73 | 67.30 | 0 |
| Genus | Stenotrophomonas | 15 | 0.0104 | 0.07 | 6.77 | 15.46 | 0 |
| Genus | Acidaminococcus | 4 | 0.1144 | 0.29 | 16.64 | 17.30 | 1 |
| Genus | WAL_1855D | 3 | 0.1397 | 0.52 | 43.59 | 43.89 | 1 |
| Genus | Oceanobacillus | 6 | 0.2589 | 0.29 | 2.06 | 3.46 | 0 |
| Family | Rikenellaceae | 38 | -0.0037 | 0.07 | 69.81 | 78.72 | 0 |
| Family | Barnesiellaceae | 25 | 0.0047 | 0.07 | 21.35 | 55.38 | 0 |
| Family | Lactobacillaceae | 99 | 0.0059 | 0.07 | 12.59 | 44.85 | 0 |
| Family | Paraprevotellaceae | 40 | 0.0104 | 0.07 | 25.70 | 38.00 | 0 |
| Family | Aeromonadaceae | 19 | 0.0126 | 0.07 | 2.87 | 3.46 | 0 |
| Order | Fusobacteriales | 65 | 0.0080 | 0.07 | 11.34 | 27.91 | 0 |
| Phylum | Cyanobacteria | 58 | 0.0129 | 0.07 | 19.81 | 52.14 | 0 |

Table 4.2: Aggregations recovered by our method at its best performance: *Aggreg. Level* is the taxonomic level at which an aggregation occurs, and *Aggregated Taxon* is the corresponding taxon at aggregation; *Size of Aggreg.* is the number of OTUs in an aggregation; $\hat{\boldsymbol{\beta}}^{\text{ours}}$ is the estimated shared coefficient for all OTUs in an aggregation; *OTU Density L.B.* is the density of the rariest OTU in an aggregation; *OTU Density U.B.* is the density of the most abundant OTU in an aggregation; *Aggreg. Density* is the density of an aggregation; *No. Shared OTUs (lasso)* is number of OTUs in an aggregation that are also selected by (4.2) at OTU level. Density is percentage of samples containing an OTU.

| OTU Index | 359105 | 182771 | 260554 | 575407 | 4306262 | 359376 | 966981 | 336632 | 192963 | 331760 | 361853 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Density (%) | 3.17 | 0.52 | 0.07 | 31.52 | 43.74 | 4.71 | 0.74 | 0.52 | 20.40 | 0.15 | 0.15 |
| $\hat{\boldsymbol{\beta}}^{\text{ours}}$ | -0.106 | -0.106 | -0.106 | -0.106 | -0.106 | -0.106 | -0.106 | -0.106 | -0.106 | -0.106 | -0.106 |
| $\hat{\boldsymbol{\beta}}^{\ell_1,\text{OTU}}$ | 0 | 0 | 0 | -0.054 | -0.083 | 0 | 0 | 0 | 0 | 0 | 0 |
| $\hat{\boldsymbol{\beta}}^{\ell_1,\text{genus}}$ | -0.133 | -0.133 | -0.133 | -0.133 | -0.133 | -0.133 | -0.133 | -0.133 | -0.133 | -0.133 | -0.133 |

Table 4.3: For the 11 OTUs associated with Akkermansia muciniphila, their densities (in percentage) and estimated regression coefficients from our method ($\hat{\boldsymbol{\beta}}^{\text{ours}}$), (4.2) at OTU level ($\hat{\boldsymbol{\beta}}^{\ell_1,\text{OTU}}$) and (4.2) at genus level ($\hat{\boldsymbol{\beta}}^{\ell_1,\text{genus}}$).

(Greer et al., 2016). The genus-level taxon Akkermansi has muciniphila as its only descendant species in the gut microbiome data. Among the 11 OTUs associated below muciniphila, only 3 of them are dense and the remaining ones are highly sparse (see Table 4.3). In addition to OTU densities, Table 4.3 also summarizes estimation by the three methods. Our method correctly aggregates all 11 OTUs into a single cluster and assigns the same regression coefficient (-0.106) to all the OTUs. As a comparison, (4.2) at OTU level only selects the two densest OTUs but sets the remaining ones to zero. Without a surprise, (4.2) at genus level also gives negative estimation (-1.333) for Akkermansi's coefficient, but its genus-level aggregation is not a data-driven decision. All three methods estimate negative effect for Akkermansia muciniphila on BMI, which is justified in previous studies.

## 4.5 Conclusion

In this chapter, we focus on regression with microbiome data and address the challenges posed by the extreme sparsity in the data. We extend the tree-based parametrization and aggregation framework proposed in Yan and Bien (2018) to compositional data setting, in which a unit-sum constraint is held. We use phylogenetic tree that relates microbes based on their similarities as side infor-

mation, and apply the tree-based aggregation framework under the log-contrast model proposed in Lin et al. (2014). We show in simulation and on a real microbiome data set that our method yield better prediction accuracy than traditional approach which arbitrarily decides a taxonomic level for aggregating OTUs. In addition, our method outputs biologically relevant aggregations for the prediction task.

A good-quality phylogenetic tree is crucial to our method's success. We show in a simulation that a distorted phylogenetic tree results in degradation of our method, where the distortion comes from missing taxonomic labels for OTUs. As an alternative, one can construct the tree by hierarchical clustering the 16S rRNA sequences with minimax linkage: a prototype can be chosen for every interior node of the tree, which improves interpretability of the tree (Bien and Tibshirani, 2011). In doing so, the new tree can grow much deeper than the phylogenetic tree that only expands across 8 taxonomic levels. The more granular hierarchical clustering tree may potentially leads to better performance for our method, as it gives more aggregation choices to the OTUs.

# CHAPTER 5

# CONCLUSIONS

We study statistical methods for getting structured sparsity patterns and structured equality patterns in parameters. In particular, we consider two scenarios for these structural patterns: (1) through hierarchical sparse modeling (HSM), for which hierarchical sparsity pattern is desired in parameters; and (2) through modeling highly sparse features that are counts of rarely occurring events, for which equality within groups of parameters is needed for appropriate aggregation of these rare features. In both cases, the methods under consideration require a tree or a DAG that encodes relations among the features as side information for achieving a desired structure in parameters.

In Chapter 2, we make a side-by-side comparison between the group lasso (GL) and latent overlapping group lasso (LOG) in HSM in terms of statistical properties and computational efficiency. We derive a closed-form solution for the proximal operator of LOG in the case of the DAG being a directed path graph. An interesting extension on that is whether a closed-form solution exists for DAG structures more general than a directed path graph. While we were not able to derive such a closed form, we have not established that such a solution does not exist. Another avenue for future work lies in extending the comparison of GL and LOG to situations beyond the class of problems considered here. For example, the sparse group lasso penalty, $\sum_{k=1}^{K} w_k \|\beta_{g_k}\|_2 + \|\beta\|_1$ (Simon et al., 2013) is a GL penalty with $K + p$ groups: $g_1, \cdots, g_K, \{1\}, \cdots, \{p\}$. This group structure can be written as $d(\mathcal{D})$, where $\mathcal{D}$ is a forest of $K$ trees, each having an empty root pointing to the singletons contained in $g_k$. However, the LOG penalty on $a(\mathcal{D})$ is simply the lasso, whereas an LOG with $g_1, \cdots, g_K, \{1\}, \cdots, \{p\}$ would seem to

be the appropriate corresponding model.

In Chapter 3, we focus on the challenge posed by highly sparse data matrices. We show, both theoretically and empirically, that not explicitly accounting for sparsity in the data hurts one's prediction errors and one's ability to perform feature selection. We propose a tree-guided parametrization and aggregation framework for modeling features counting frequency of rarely occurring events. In Chapter 3 and Chapter 4, we apply our method to the hotel review data from TripAdvisor and the microbiome abundance data from the American Gut project. While both applications target on continuous numerical variables, such as rating score and BMI, it would be interesting to extend our feature aggregation framework to a classification setting. Another important extension is developing more efficient algorithm for our proposed method. The current ADMM algorithm for our method requires singular value decomposition (SVD) in the dimension of $p$. In high dimensions when $p$ is very large, SVD can be computationally intensive and the following updates can be costly and slow. A stochastic-type algorithm that does not require SVD over all $p$ features is preferred for our method.

APPENDIX A

**APPENDIX FOR CHAPTER 2**

## A.1 Proof of Lemma 1

For $p = 2$, denote $\beta = (\beta_1, \beta_2) \in \mathbb{R}^2$. The $\Omega_{\text{GL}}^{d(\mathcal{D})}$ and $\Omega_{\text{LOG}}^{a(\mathcal{D})}$ penalties can be written as

$$\Omega_{\text{GL}}^{d(\mathcal{D})}(\beta; w) = w_1 \|(\beta_1, \beta_2)\|_2 + w_2 |\beta_2|,$$

$$\Omega_{\text{LOG}}^{a(\mathcal{D})}(\beta; w') = \min_{\{v_1^{(1)} \in \mathbb{R}, v^{(2)} \in \mathbb{R}^2\}} \left\{ |v_1^{(1)}| + \|v^{(2)}\|_2 \quad \text{s.t.} \quad \begin{pmatrix} v_1^{(1)} + v_1^{(2)} \\ v_2^{(2)} \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} \right\}.$$

Suppose there exists $w \in \mathbb{R}^{+2}$ such that for all $\beta$, $\Omega_{\text{GL}}^{d(\mathcal{D})}(\beta; w) = \Omega_{\text{LOG}}^{a(\mathcal{D})}(\beta; w')$ holds. The equality also holds for $\beta = (0, \beta_2)$ and $\beta = (\beta_1, 0)$.

- When $\beta = (0, \beta_2)$, i.e. $\beta_1 = 0$, the following is true

  $$\Omega_{\text{LOG}}^{a(\mathcal{D})}(\beta; w') = \min_{v_1^{(1)} \in \mathbb{R}} |v_1^{(1)}| + \sqrt{(v_1^{(1)})^2 + \beta_2^2} = |\beta_2| = \Omega_{\text{GL}}^{d(\mathcal{D})}(\beta; w) = (w_1 + w_2)|\beta_2|.$$

  We get $w_1 + w_2 = 1$.

- When $\beta = (\beta_1, 0)$, i.e. $\beta_2 = 0$, the following is true

  $$\Omega_{\text{LOG}}^{a(\mathcal{D})}(\beta; w') = \min_{v_1^{(2)} \in \mathbb{R}} |\beta_1 - v_1^{(2)}| + |v_1^{(2)}| = |\beta_1| = \Omega_{\text{GL}}^{d(\mathcal{D})}(\beta; w) = w_1 |\beta_1|.$$

  We get $w_1 = 1$.

Combining the results above we have $w_2 = 0$ which leads to a contradiction. Hence, when $p = 2$ and $w' = (1, 1)$, there does not exist $w \in \mathbb{R}^{+2}$ such that $\Omega_{\text{GL}}^{d(\mathcal{D})}(\cdot; w) = \Omega_{\text{LOG}}^{a(\mathcal{D})}(\cdot; w')$.

## A.2 Proof of Propositions 1 and 2

Let $y = \beta^* + \epsilon$ where $\epsilon \sim N_D(0, \sigma^2 I_D)$ and $\beta^*_d = 1_{\{d \le K^*\}}$ for $d = 1, \ldots, D$ (and assume $K^* < D$). We define the event

$$\mathcal{B} := \left\{ \max_{i=1,\ldots,D} |\epsilon_i| > \bar{\lambda} \right\} \tag{A.1}$$

where $\bar{\lambda} := 2\sigma\sqrt{\log D}$. A union bound and a Chernoff upper bound for normal variables establishes that

$$\mathbb{P}(\max_{i=1,\ldots,D} |\epsilon_i| > t) \le D\mathbb{P}(|\epsilon_1| > t) \le 2De^{-t^2/2\sigma^2},$$

for $t > 0$. Taking $t = 2\sigma\sqrt{\log D}$ gives that

$$\mathbb{P}(\mathcal{B}) \le 2/D. \tag{A.2}$$

### A.2.1 Proof of Proposition 1

We establish the following deterministic result that holds on $\mathcal{B}^c$ (by (A.2), this proves Proposition 1).

**Lemma 7.** *The following two statements hold on $\mathcal{B}^c$ under the assumptions of Proposition 1:*

*(a)* $supp(\hat{\beta}^{\mathrm{GL}}) \subseteq supp(\beta^*)$

*(b) For* $1 \le d \le d + h \le K^*$ *and* $\hat{\beta}^{\mathrm{GL}}_d \neq 0$,

$$\frac{\left|\hat{\beta}^{\mathrm{GL}}_{d+h}\right|}{\left|\hat{\beta}^{\mathrm{GL}}_{d}\right|} \le \frac{|y_{d+h}|}{|y_d|} \exp\left( - \frac{\lambda h}{\sqrt{\sum_{m=d+1}^{K^*} y_m^2}} \right).$$

*Proof.* Jenatton et al. (2011b) provide a closed-form solution for (2.9) (see Algorithm 2 in their paper). Their algorithm in this context is as follows:

1. Initialize $\hat{b}^{(D)} = y$.
2. For $d = D, \ldots, 1$,

$$\hat{b}^{(d-1)}_{d:D} \leftarrow \hat{b}^{(d)}_{d:D} \cdot \left(1 - \frac{\lambda}{\left\|\hat{b}^{(d)}_{d:D}\right\|_2}\right)_+, \text{ and } \quad \hat{b}^{(d-1)}_{1:(d-1)} \leftarrow y_{1:(d-1)} \text{ if } d > 1.$$

Defining $\hat{r}_d := \left\|\hat{b}^{(d)}_{d:D}\right\|_2$ for $d = 1, \ldots, D$, one gets the recurrence relation

$$\hat{r}^2_{d-1} = (\hat{r}_d - \lambda)^2_+ + y^2_{d-1} \text{ where } \hat{r}_D = |y_D|. \tag{A.3}$$

The solution to (2.9) can be expressed, for each $d$, as

$$\hat{\beta}^{\mathrm{GL}}_d = y_d \cdot \prod_{\ell=1}^{d} (1 - \lambda/\hat{r}_\ell)_+ . \tag{A.4}$$

Our choice of $\lambda$ in Proposition 1 establishes that on $\mathcal{B}^c$, $\lambda > \max_{i=1,\ldots,D} |\epsilon_i|$. This together with the recurrence relation in (A.3) implies the following:

- For $d = K^* + 1, \ldots, D$, $\hat{r}_d = |y_d| = |\epsilon_d|$ and thus, by (A.4), $\hat{\beta}^{\mathrm{GL}}_d = 0$ for $d > K^*$. This establishes that $\mathrm{supp}(\hat{\beta}^{\mathrm{GL}}) \subseteq \mathrm{supp}(\beta^*)$.
- For $d = K^*$,

$$\hat{r}_{K^*} = |y_{K^*}| \leq \sqrt{\sum_{\ell=d}^{K^*} y^2_\ell}.$$

  For $d < K^*$, suppose $\hat{r}_{d+1} \leq \sqrt{\sum_{\ell=d+1}^{K^*} y^2_\ell}$. Then we have

$$\hat{r}_d = \sqrt{(\hat{r}_{d+1} - \lambda)^2_+ + y^2_d} \leq \sqrt{\hat{r}^2_{d+1} + y^2_d} \leq \sqrt{\sum_{\ell=d+1}^{K^*} y^2_\ell + y^2_d} \leq \sqrt{\sum_{\ell=d}^{K^*} y^2_\ell}.$$

  This establishes by induction that $\hat{r}_d \leq \sqrt{\sum_{\ell=d}^{K^*} y^2_\ell}$ for all $d \leq K^*$.

  For $1 \leq d \leq d + h \leq K^*$, assuming $\hat{\beta}^{\mathrm{GL}}_d \neq 0$, we have

$$\frac{\left|\hat{\beta}^{\mathrm{GL}}_{d+h}\right|}{\left|\hat{\beta}^{\mathrm{GL}}_d\right|} = \frac{|y_{d+h}| \cdot \prod_{\ell=1}^{d+h} \left(1 - \frac{\lambda}{\hat{r}_\ell}\right)_+}{|y_d| \cdot \prod_{\ell=1}^{d} \left(1 - \frac{\lambda}{\hat{r}_\ell}\right)_+}$$

$$
\begin{aligned}
&= \frac{|y_{d+h}|}{|y_d|} \prod_{\ell=d+1}^{d+h} \left(1 - \frac{\lambda}{\hat{r}_\ell}\right)_+ \\
&\leq \frac{|y_{d+h}|}{|y_d|} \exp\left(-\sum_{\ell=d+1}^{d+h} \frac{\lambda}{\hat{r}_\ell}\right) \quad (\text{since } (1-x)_+ \leq e^{-x} \text{ for } x \in \mathbb{R}) \\
&\leq \frac{|y_{d+h}|}{|y_d|} \exp\left(-\sum_{\ell=d+1}^{d+h} \frac{\lambda}{\sqrt{\sum_{m=\ell}^{K^*} y_m^2}}\right) \\
&\leq \frac{|y_{d+h}|}{|y_d|} \exp\left(-\frac{\lambda h}{\sqrt{\sum_{m=d+1}^{K^*} y_m^2}}\right).
\end{aligned}
$$

□

### A.2.2 Proof of Proposition 2

We prove two deterministic lemmas, corresponding to parts (a) and (b) in Proposition 2.

**Lemma 8.** *Under the assumptions of Proposition 2, $supp(\hat{\beta}^{\text{LOG}}) \subseteq supp(\beta^*)$ holds on $\mathcal{B}^c$.*

*Proof.* We prove this using Algorithm 3 which solves (2.10) under a directed path graph. Let $\bar{K}$ be the largest knot such that $\bar{K} \leq K^*$, determined by Algorithm 3 on solving (2.10). We show in what follows that $f(k, \bar{K}) \leq \lambda \; \forall \; k > \bar{K}$, which establishes that $\bar{K}$ is the last knot. The assumed lower bound on $\lambda$ ensures that $\lambda > \max_{i=1,\ldots,D} |\epsilon_i|$ on the event $\mathcal{B}^c$.

If $\bar{K} = K^*, \forall \; k > K^*$

$$
f(k, \bar{K}) = \frac{\left\|y_{(\bar{K}+1):k}\right\|_2}{\sqrt{k - \bar{K}}} = \frac{\left\|\epsilon_{(K^*+1):k}\right\|_2}{\sqrt{k - K^*}} \leq \max_{i=1,\ldots,D} |\epsilon_i| < \lambda. \tag{A.5}
$$

If $\bar{K} < K^*$, then $K^*$ is not chosen as a knot, given that $\bar{K}$ by construction is the last knot determined by Algorithm 3 on or before $K^*$. We let $k = \bar{K}$ on line 5 of Algorithm 3. Consider two possible cases for $K^*$: $K^* = \arg\max_{j:j>\bar{K}} f(j, \bar{K})$ and $K^* \neq \arg\max_{j:j>\bar{K}} f(j, \bar{K})$. In the first case, we must have $f(K^*, \bar{K}) \leq \lambda$ otherwise the while loop would not break on line 7, making $K^*$ a knot and leading to a contradiction. In the second case, let $\check{K} := \arg\max_{j:j>\bar{K}} f(j, \bar{K})$. If $\check{K} < K^*$, we have $f(K^*, \bar{K}) \leq f(\check{K}, \bar{K}) \leq \lambda$ otherwise $\check{K}$ would be a knot which would then be in contradiction with the assumption that $\bar{K}$ was the last knot on or before $K^*$. If $\check{K} > K^*$, we have $f(K^*, \bar{K}) \leq f(\check{K}, \bar{K})$ by definition of $\check{K}$. In summary, either one of the following is true for the second case:

(i) $f(K^*, \bar{K}) \leq \lambda$

(ii) $\exists\, \bar{k} > K^*$ such that $f(K^*, \bar{K}) \leq f(\bar{k}, \bar{K})$, i.e., $\left\|y_{(\bar{K}+1):K^*}\right\|_2^2 \leq \left\|y_{(\bar{K}+1):\bar{k}}\right\|_2^2 \cdot \frac{K^*-\bar{K}}{\bar{k}-\bar{K}}$.

We show that in both cases

$$\left\|y_{(\bar{K}+1):K^*}\right\|_2^2 \leq \lambda^2(K^* - \bar{K}). \tag{A.6}$$

Case (i) is equivalent to (A.6). When Case (ii) holds, $\exists\, \bar{k} > K^*$ such that

$$\left\|y_{(\bar{K}+1):K^*}\right\|_2^2 \leq \left\|y_{(\bar{K}+1):\bar{k}}\right\|_2^2 \cdot \frac{K^* - \bar{K}}{\bar{k} - \bar{K}} = \left(\left\|y_{(\bar{K}+1):K^*}\right\|_2^2 + \left\|\epsilon_{(K^*+1):\bar{k}}\right\|_2^2\right) \cdot \frac{K^* - \bar{K}}{\bar{k} - \bar{K}}. \tag{A.7}$$

Plugging $\alpha = \frac{K^*-\bar{K}}{\bar{k}-\bar{K}}$ into (A.7) yields

$$\begin{aligned} & (1-\alpha)\left\|y_{(\bar{K}+1):K^*}\right\|_2^2 \leq \alpha \left\|\epsilon_{(K^*+1):\bar{k}}\right\|_2^2 \\ \Rightarrow \quad & \left\|y_{(\bar{K}+1):K^*}\right\|_2^2 \leq \frac{\alpha}{1-\alpha}\left\|\epsilon_{(K^*+1):\bar{k}}\right\|_2^2 < \frac{\alpha}{1-\alpha}\lambda^2(\bar{k} - K^*) = \lambda^2(K^* - \bar{K}) \end{aligned}$$

where the last equality is by $\frac{\alpha}{1-\alpha}(\bar{k} - K^*) = K^* - \bar{K}$. Having established that (A.6) holds, we have $\forall\, k > K^*$ that

$$\left\|y_{(\bar{K}+1):k}\right\|_2^2 = \left\|y_{(\bar{K}+1):K^*}\right\|_2^2 + \left\|\epsilon_{(K^*+1):k}\right\|_2^2 < \lambda^2(K^* - \bar{K}) + \lambda^2(k - K^*) = \lambda^2(k - \bar{K}) \tag{A.8}$$

By (A.8) we have

$$\left\|y_{(\bar{K}+1):k}\right\|_2^2 \leq \lambda^2(k-\bar{K}) \quad \Leftrightarrow \quad f(k,\bar{K}) \leq \lambda. \tag{A.9}$$

According to Algorithm 3, $\bar{K}$ is the last knot on the entire path graph and $\text{supp}(\hat{\beta}^{\text{LOG}}) \subseteq \text{supp}(\beta^*)$. □

**Lemma 9.** *Under the assumptions of Proposition 2, the following holds on $\mathcal{B}^c$: For $1 \leq d \leq d+h \leq K^*$ and $\hat{\beta}^{\text{LOG}}_{d+h} \neq 0$,*

$$\delta \frac{|y_{d+h}|}{|y_d|} \leq \frac{\left|\hat{\beta}^{\text{LOG}}_{d+h}\right|}{\left|\hat{\beta}^{\text{LOG}}_{d}\right|} \leq \frac{|y_{d+h}|}{|y_d|}.$$

*Proof.* For $1 \leq d \leq d+h \leq K^*$ and $\hat{\beta}^{\text{LOG}}_{d+h} \neq 0$, by Algorithm 3 we have

$$\frac{\left|\hat{\beta}^{\text{LOG}}_{d+h}\right|}{\left|\hat{\beta}^{\text{LOG}}_{d}\right|} = \frac{|y_{d+h}|}{|y_d|} \cdot \frac{1-\frac{\lambda}{f(k^U(d+h),k^L(d+h))}}{1-\frac{\lambda}{f(k^U(d),k^L(d))}} \tag{A.10}$$

where $f(k,j) = \|y_{(j+1):k}\|_2/\sqrt{k-j}$ and $k^L(d)$ and $k^U(d)$ are two adjacent knots determined by Algorithm 3 such that $k^L(d) < d \leq k^U(d)$ (and similarly $k^L(d+h) < d+h \leq k^U(d+h)$). For simplicity of notation, we denote $a := f(k^U(d+h), k^L(d+h))$ and $b := f(k^U(d), k^L(d))$.

By (A.10), we wish to show that

$$\delta \leq \frac{1-\lambda/a}{1-\lambda/b} \leq 1.$$

When $k^L(d) = k^L(d+h)$ and $k^U(d) = k^U(d+h)$, $a = b$ and thus this is immediate. It remains to consider the case when $k^L(d) < k^U(d) \leq k^L(d+h) < k^U(d+h)$. By Lemma 10, we have that $b \geq a$, which gives the upper bound.

Some algebra shows that

$$\frac{1-\lambda/a}{1-\lambda/b} \geq \delta \iff \lambda \leq \frac{1-\delta}{1/a-\delta/b}. \tag{A.11}$$

We will show that the upper bound on $\lambda$ assumed in Proposition 2 ensures that the above inequality holds on $\mathcal{B}^c$.

For any $0 \le j < k \le K^*$,

$$\min_{i \in \{j+1,\dots,k\}} y_i^2 \le \|y_{(j+1):k}\|_2^2/(k-j) = f(k,j)^2$$

and thus on $\mathcal{B}^c$,

$$\begin{aligned} f(k,j) \ge \min_{1 \le i \le K^*} |y_i| &\ge 1 - \max_{1 \le i \le K^*} |\epsilon_i| \qquad \text{by the triangle inequality} \\ &\ge 1 - \bar{\lambda} \qquad \text{by definition of } \mathcal{B}^c. \end{aligned} \tag{A.12}$$

Since $supp(\hat{\beta}^{\text{LOG}}) \subseteq supp(\beta^*)$ on $\mathcal{B}^c$ by Lemma 8 and $\hat{\beta}^{\text{LOG}}_{d+h} \neq 0$ by assumption, we have $k^L(d+h) < k^U(d+h) \le K^*$. Taking $(k,j) = (k^U(d+h), k^L(d+h))$ in (A.12) yields

$$1 - \bar{\lambda} \le a \le \frac{1}{1/a - \delta/b}.$$

Thus, recalling the upper bound for $\lambda$ given in Proposition 2,

$$\lambda \le (1-\delta)(1-\bar{\lambda}) \le \frac{1-\delta}{1/a - \delta/b},$$

which by (A.11), establishes that

$$\frac{\left|\hat{\beta}^{\text{LOG}}_{d+h}\right|}{\left|\hat{\beta}^{\text{LOG}}_{d}\right|} \ge \frac{|y_{d+h}|}{|y_d|} \cdot \delta.$$

□

## A.3 Proof that Algorithm 3 Solves $\text{Prox}^{a(\mathcal{D})}_{\text{LOG}}$ for a Directed Path Graph

Suppose $\mathcal{D}$ is a directed path graph with $D$ nodes as shown in Figure 2.3. Let $\hat{\beta} = \text{Prox}^{a(\mathcal{D})}_{\text{LOG}}(y; \lambda', w')$ and $\bar{\beta}$ denote the output from Algorithm 3 with inputs $\lambda'$

and $w'$. To prove $\bar{\beta} = \hat{\beta}$, we propose a $\{\bar{v}^{(\ell)}\}_{\ell=1}^{D}$ such that $\mathrm{supp}(\bar{v}^{(\ell)}) \subseteq s_{1:\ell}$ and $\bar{v}^{(\ell)} \in \mathbb{R}^p$ for $\ell = 1, \ldots, D$. We then show that $\bar{\beta} = \sum_{\ell=1}^{D} \bar{v}^{(\ell)}$ and

$$\begin{cases} \bar{\beta}_{s_{1:\ell}} - y_{s_{1:\ell}} = -\frac{\lambda' w'_\ell \bar{v}^{(\ell)}}{\|\bar{v}^{(\ell)}\|_2} & \text{if } \bar{v}^{(\ell)} \neq 0 \\ \|\bar{\beta}_{s_{1:\ell}} - y_{s_{1:\ell}}\|_2 \leq \lambda' w'_\ell & \text{if } \bar{v}^{(\ell)} = 0. \end{cases} \tag{A.13}$$

By the optimality condition stated in Lemma 11 of Obozinski et al. (2011), this establishes that $\bar{\beta} = \hat{\beta}$. Let $0 = k_0 < k_1 < \cdots < k_m \leq D$ be the sequence of knots determined by Algorithm 3 such that $k_i$ maximizes $f(\cdot, k_{i-1})$ and $f(k_i, k_{i-1}) > \lambda'$ for $i = 1, \ldots, m$.

If $m = 0$, i.e., $k_0 = 0$ is the only knot, we have $\bar{\beta} = 0$. Consider $\bar{v}^{(\ell)} = 0$ for $\ell = 1, \ldots, D$, which satisfy $\bar{\beta} = \sum_{\ell=1}^{D} \bar{v}^{(\ell)}$. Moreover, we get $\|y_{s_{1:\ell}}\|_2 / w'_\ell \leq \lambda'$ for $\ell = 1, \ldots, D$ directly from the algorithm. By Lemma 11 of Obozinski et al. (2011), $\bar{\beta} = \hat{\beta}$.

Now consider $m \geq 1$. We first prove an inequality in $f(j, k)$ in Lemma 10 when $(k, j)$ are two nearest knots.

**Lemma 10.** *Let $0 = k_0 < k_1 < \cdots < k_m \leq D$ be the sequence of knots. We have the following inequality.*

$$f(k_{j-1}, k_{j-2}) \geq f(k_j, k_{j-1}), \quad \textit{for } j = 2, \ldots, m.$$

*Proof.* Applying Algorithm 3 yields that for $j = 2, \ldots, m$,

$$\begin{aligned} & f(k_{j-1}, k_{j-2}) \geq f(k_j, k_{j-2}) \\ \Rightarrow \quad & \frac{\|y_{s_{(k_{j-2}+1):k_{j-1}}}\|_2}{\sqrt{w'^2_{k_{j-1}} - w'^2_{k_{j-2}}}} \geq \frac{\|y_{s_{(k_{j-2}+1):k_j}}\|_2}{\sqrt{w'^2_{k_j} - w'^2_{k_{j-2}}}} \\ \Rightarrow \quad & \frac{w'^2_{k_{j-1}} - w'^2_{k_{j-2}}}{\|y_{s_{(k_{j-2}+1):k_{j-1}}}\|_2^2} \leq \frac{w'^2_{k_j} - w'^2_{k_{j-2}}}{\|y_{s_{(k_{j-2}+1):k_j}}\|_2^2} \\ \Rightarrow \quad & \frac{w'^2_{k_{j-1}} - w'^2_{k_{j-2}}}{\|y_{s_{(k_{j-2}+1):k_{j-1}}}\|_2^2} - \frac{w'^2_{k_{j-1}} - w'^2_{k_{j-2}}}{\|y_{s_{(k_{j-2}+1):k_j}}\|_2^2} \leq \frac{w'^2_{k_j} - w'^2_{k_{j-2}}}{\|y_{s_{(k_{j-2}+1):k_j}}\|_2^2} - \frac{w'^2_{k_{j-1}} - w'^2_{k_{j-2}}}{\|y_{s_{(k_{j-2}+1):k_j}}\|_2^2} \end{aligned}$$

$$
\begin{aligned}
\Rightarrow \quad & \frac{(w'^2_{k_{j-1}} - w'^2_{k_{j-2}})\|y_{s_{(k_{j-1}+1):k_j}}\|_2^2}{\|y_{s_{(k_{j-2}+1):k_{j-1}}}\|_2^2 \|y_{s_{(k_{j-2}+1):k_j}}\|_2^2} \leq \frac{w'^2_{k_j} - w'^2_{k_{j-1}}}{\|y_{s_{(k_{j-2}+1):k_j}}\|_2^2} \\
\Rightarrow \quad & \frac{w'^2_{k_{j-1}} - w'^2_{k_{j-2}}}{\|y_{s_{(k_{j-2}+1):k_{j-1}}}\|_2^2} \leq \frac{w'^2_{k_j} - w'^2_{k_{j-1}}}{\|y_{s_{(k_{j-1}+1):k_j}}\|_2^2} \\
\Rightarrow \quad & \frac{\sqrt{w'^2_{k_{j-1}} - w'^2_{k_{j-2}}}}{\|y_{s_{(k_{j-2}+1):k_{j-1}}}\|_2} \leq \frac{\sqrt{w'^2_{k_j} - w'^2_{k_{j-1}}}}{\|y_{s_{(k_{j-1}+1):k_j}}\|_2} \\
\Rightarrow \quad & \frac{1}{f(k_{j-1}, k_{j-2})} \leq \frac{1}{f(k_j, k_{j-1})} \\
\Rightarrow \quad & f(k_{j-1}, k_{j-2}) \geq f(k_j, k_{j-1})
\end{aligned}
$$

□

For notational simplicity, we let $a_j = f(k_j, k_{j-1})$ for $j = 1, \ldots, m$, and let

$$A_j = \sum_{i=1}^{j} \frac{y_{s_{(k_{i-1}+1):k_i}}}{a_i}.$$

We observe that

$$\|A_j\|_2^2 = \sum_{i=1}^{j} \frac{\|y_{s_{(k_{i-1}+1):k_i}}\|_2^2}{a_i^2} = \sum_{i=1}^{j} (w'^2_{k_i} - w'^2_{k_{i-1}}) = w'^2_{k_j}. \tag{A.14}$$

Now consider the following $\{\bar{v}^{(\ell)}\}_{\ell=1}^{D}$ such that $\text{supp}(\bar{v}^{(\ell)}) \subseteq s_{1:\ell}$ and $\bar{v}^{(\ell)} \in \mathbb{R}^p \ \forall \ell$.

- For $\ell \notin \{k_1, \cdots, k_m\}$,

$$\bar{v}^{(\ell)} = 0.$$

- For $\ell = k_j$ for $j = 1, \ldots, m-1$,

$$
\begin{aligned}
\bar{v}^{(k_j)} &= S_G\left(a_j A_j, \quad w'_{k_j} a_{j+1}\right) \\
&= A_j \cdot a_j \cdot \left(1 - \frac{w'_{k_j} a_{j+1}}{a_j \|A_j\|_2}\right)_+ \\
&= A_j \cdot \left(a_j - a_{j+1}\right)
\end{aligned}
$$

by (A.14) and $a_j \geq a_{j+1}$ from Lemma 10.

- For $\ell = k_m$,

$$\begin{aligned}\bar{v}^{(k_m)} &= S_G\left(a_m A_m, \quad \lambda' w'_{k_m}\right) \\ &= a_m A_m \cdot \left(1 - \frac{\lambda' w'_{k_m}}{a_m ||A_m||_2}\right)_+ \\ &= A_m \cdot (a_m - \lambda')\end{aligned}$$

  by $a_m > \lambda'$ from Algorithm 3.

Because of the very definition of $\bar{\beta}$ in Algorithm 3, we can express $\bar{\beta}$ in the following form:

- For $1 \le i \le m$, $\bar{\beta}_{s_{(k_{i-1}+1):k_i}} = S_G\left(y_{s_{(k_{i-1}+1):k_i}}, \lambda' \sqrt{w'^2_{k_i} - w'^2_{k_{i-1}}}\right)$.
- If $k_m < D$, $\bar{\beta}_{s_{(k_m+1):D}} = 0$.

We show that $\bar{\beta} = \sum_{\ell=1}^{D} \bar{v}^{(\ell)}$ through steps $(a)$, $(b)$ and $(c)$ below.

$(a)$ For $i = 1, \ldots, m-1$,

$$\begin{aligned}\sum_{\ell=1}^{D} \bar{v}^{(\ell)}_{s_{(k_{i-1}+1):k_i}} &= \sum_{j=i}^{m} \bar{v}^{(k_j)}_{s_{(k_{i-1}+1):k_i}} \\ &= \sum_{j=i}^{m-1} \bar{v}^{(k_j)}_{s_{(k_{i-1}+1):k_i}} + \bar{v}^{(k_m)}_{s_{(k_{i-1}+1):k_i}} \\ &= \frac{y_{s_{(k_{i-1}+1):k_i}}}{a_i} \sum_{j=i}^{m-1} \left(a_j - a_{j+1}\right) + \frac{y_{s_{(k_{i-1}+1):k_i}}}{a_i} (a_m - \lambda') \\ &= \frac{y_{s_{(k_{i-1}+1):k_i}}}{a_i} (a_i - \lambda') \\ &= y_{s_{(k_{i-1}+1):k_i}} \left(1 - \frac{\lambda'}{a_i}\right) \\ &= S_G\left(y_{s_{(k_{i-1}+1):k_i}}, \lambda' \sqrt{w'^2_{k_i} - w'^2_{k_{i-1}}}\right) = \bar{\beta}_{s_{(k_{i-1}+1):k_i}}.\end{aligned}$$

(*b*) For $i = m$,

$$\begin{aligned}
\sum_{\ell=1}^{D} \bar{v}^{(\ell)}_{s_{(k_{i-1}+1):k_i}} &= \bar{v}^{(k_m)}_{s_{(k_{m-1}+1):k_m}} \\
&= \frac{y_{s_{(k_{m-1}+1):k_m}}}{a_m} (a_m - \lambda') \\
&= y_{s_{(k_{m-1}+1):k_m}} \left(1 - \frac{\lambda'}{a_m}\right) \\
&= S_G \left(y_{s_{(k_{m-1}+1):k_m}}, \lambda' \sqrt{w'^2_{k_m} - w'^2_{k_{m-1}}}\right) \\
&= \bar{\beta}_{s_{(k_{m-1}+1):k_m}}.
\end{aligned}$$

(*c*) If $k_m < D$, $\sum_{\ell=1}^{D} \bar{v}^{(\ell)}_{s_{(k_m+1):D}} = 0 = \bar{\beta}_{s_{(k_m+1):D}}$.

Combining (*a*), (*b*) and (*c*) we have established $\bar{\beta} = \sum_{\ell=1}^{D} \bar{v}^{(\ell)}$. We next show (A.13) is true through steps (*a*′) and (*b*′) below.

(*a*′) By definition, $\bar{v}^{(\ell)} \neq 0$ if and only if $\ell \in \{k_1, \cdots, k_m\}$. For $\ell = k_i \in \{k_1, \cdots, k_m\}$, we have

$$\begin{aligned}
\bar{\beta}_{s_{1:k_i}} - y_{s_{1:k_i}} &= \sum_{j=1}^{i} S_G \left(y_{s_{(k_{j-1}+1):k_j}}, \lambda' \sqrt{w'^2_{k_j} - w'^2_{k_{j-1}}}\right) - y_{s_{1:k_i}} \\
&= \sum_{j=1}^{i} y_{s_{(k_{j-1}+1):k_j}} (1 - \lambda' a_j^{-1}) - y_{s_{1:k_i}} \\
&= \sum_{j=1}^{i} -\frac{\lambda' y_{s_{(k_{j-1}+1):k_j}}}{a_j} = -\lambda' A_i.
\end{aligned}$$

By the definition of $\{\bar{v}^{(\ell)}\}_{\ell=1}^{D}$, we have

$$-\frac{\lambda' w'_{k_i} \bar{v}^{(k_i)}}{\|\bar{v}^{(k_i)}\|_2} = -\frac{\lambda' w'_{k_i} A_i}{\|A_i\|_2} = -\lambda' A_i.$$

Thus, $\bar{\beta}_{s_{1:\ell}} - y_{s_{1:\ell}} = -\frac{\lambda' w'_\ell \bar{v}^{(\ell)}}{\|\bar{v}^{(\ell)}\|_2}$ if $\bar{v}^{(\ell)} \neq 0$.

(*b*′) By definition, $\bar{v}^{(\ell)} = 0$ if and only if $\ell \notin \{k_1, \cdots, k_m\}$. We discuss $\ell$ in the following three cases.

(i) If $k_{i-1} < \ell < k_i$ for some $i = 2, \ldots, m$, by Algorithm 3 we have

$$\bar{\beta}_{s_{1:\ell}} - y_{s_{1:\ell}} = -\lambda' A_{i-1} - \frac{\lambda' y_{s_{(k_{i-1}+1):\ell}}}{a_i}.$$

Taking $\ell_2$-norm on both sides yields

$$\|\bar{\beta}_{s_{1:\ell}} - y_{s_{1:\ell}}\|_2 = \lambda' \sqrt{w'^2_{k_{i-1}} + \frac{(w'^2_{k_i} - w'^2_{k_{i-1}})\|y_{s_{(k_{i-1}+1):\ell}}\|_2^2}{\|y_{s_{(k_{i-1}+1):k_i}}\|_2^2}} \tag{A.15}$$

By the Algorithm 3, we know

$$k_i = \underset{i' \in \{k_{i-1}+1, \cdots, D\}}{\arg\max} f(i', k_{i-1}),$$

so that $a_i \geq f(\ell, k_{i-1})$ which leads to

$$\frac{\|y_{s_{(k_{i-1}+1):k_i}}\|_2^2}{(w'^2_{k_i} - w'^2_{k_{i-1}})} \geq \frac{\|y_{s_{(k_{i-1}+1):\ell}}\|_2^2}{(w'^2_{\ell} - w'^2_{k_{i-1}})} \quad \Rightarrow \quad \frac{(w'^2_{k_i} - w'^2_{k_{i-1}})\|y_{s_{(k_{i-1}+1):\ell}}\|_2^2}{\|y_{s_{(k_{i-1}+1):k_i}}\|_2^2} \leq w'^2_{\ell} - w'^2_{k_{i-1}}. \tag{A.16}$$

Combining (A.15) and (A.16) yields $\|\bar{\beta}_{s_{1:\ell}} - y_{s_{1:\ell}}\|_2 \leq \lambda' \sqrt{w'^2_{k_{i-1}} + w'^2_{\ell} - w'^2_{k_{i-1}}} = \lambda' w'_{\ell}$.

(ii) If $\ell < k_1$, $\bar{\beta}_{s_{1:\ell}} - y_{s_{1:\ell}} = -\lambda' y_{s_{1:\ell}}/a_1$. Since $k_1 = \arg\max_{i' \in \{1,\ldots,D\}} f(i', 0)$, we have $a_1 \geq f(\ell, 0)$ which leads to

$$\frac{\|y_{s_{1:k_1}}\|_2^2}{w'^2_{k_1}} \geq \frac{\|y_{s_{1:\ell}}\|_2'^2}{w'^2_{\ell}} \quad \Rightarrow \quad \frac{w'^2_{k_1}\|y_{s_{1:\ell}}\|_2^2}{\|y_{s_{1:k_1}}\|_2^2} \leq w'^2_{\ell}. \tag{A.17}$$

By (A.17) we get

$$\|\bar{\beta}_{s_{1:\ell}} - y_{s_{1:\ell}}\|_2 = \sqrt{\frac{\lambda'^2 w'^2_{k_1}\|y_{s_{1:\ell}}\|_2^2}{\|y_{s_{1:k_1}}\|_2^2}} \leq \lambda' w'_{\ell}.$$

(iii) If $\ell > k_m$ (provided $k_m < D$),

$$\bar{\beta}_{s_{1:\ell}} - y_{s_{1:\ell}} = -\lambda' A_m - y_{s_{(k_m+1):\ell}}$$

Since $k_m$ is the last knot, we know that

$$\max_{i' \in \{k_m+1, \cdots, D\}} f(i', k_m) \leq \lambda'.$$

Thus, $f(\ell, k_m) \leq \lambda'$ which leads to

$$\|y_{s_{(k_m+1):\ell}}\|_2^2 \leq \lambda'^2(w'^2_\ell - w'^2_{k_m}).$$

Thus,

$$\begin{aligned}
\|\bar{\beta}_{s_{1:\ell}} - y_{s_{1:\ell}}\|_2 &= \sqrt{\lambda'^2\|A_m\|_2^2 + \|y_{s_{(k_m+1):\ell}}\|_2^2} \\
&= \sqrt{\lambda'^2 w'^2_{k_m} + \|y_{s_{(k_m+1):\ell}}\|_2^2} \\
&\leq \sqrt{\lambda'^2 w'^2_{k_m} + \lambda'^2(w'^2_\ell - w'^2_{k_m})} = \lambda' w'_\ell.
\end{aligned}$$

Combining $(a')$ and $(b')$ we prove (A.13) holds. Since the second optimality condition in Lemma 11 of Obozinski et al. (2011) is satisfied, we have $\bar{\beta} = \hat{\beta}$.

## A.4 Computational Complexity of Algorithm 3

Let $z_i = \|y_{s_i}\|_2^2$ for $i = 1, \ldots, D$. We begin by computing all the $z_i$, which takes $O(p)$ operations. To compute the $i$th knot requires computing $f(j, k_{i-1})$ for $j = k_{i-1} + 1, \ldots, D$.

To compute $f(k+1, k)^2 = z_{k+1}/(w_{k+1}^2 - w_k^2))$ requires constant time; also, once $f(j, k)$ has been computed, we can get $f(j+1, k)$ in constant time since

$$f(j+1, k)^2 = \frac{(w_j^2 - w_k^2)f(j, k)^2 + z_{j+1}}{(w_{j+1}^2 - w_k^2)}.$$

Thus computing all the $f(\cdot, k_{i-1})$'s requires $O(D - k_{i-1})$ operations. Finding the maximizer in line 5 takes an additional $O(D - k_{i-1})$ operations. Thus, in total finding all knots requires on the order of

$$p + \sum_{i=1}^{m}(D - k_{i-1})$$

operations. Once the knots have been found, the groupwise soft-thresholding steps require only an additional $O(p)$ work. Therefore, the algorithm requires $O(p+mD)$ operations. Since the number of knots is not known *a priori*, the worst case is $O(p + D^2)$.

## A.5 Computational Complexity of GL for a Directed Path Graph

### A.5.1 GL Proximal Operator

By Jenatton et al. (2011b)'s result, Algorithm 1 will converge in a single pass when $\mathcal{D}$ is a directed path graph if we cycle through the groups $g_i = s_{(D+1-i):D}$ from smallest to largest. The algorithm can be stated simply as follows: Initialize $\beta^0 = y$ and then for $i = 1, \ldots, D$, set

$$\beta^i_{g_i} \leftarrow \left(1 - \frac{\lambda w_i}{\|\beta^{i-1}_{g_i}\|_2}\right)_+ \beta^{i-1}_{g_i},$$

and output $\beta^D$ as the solution. As in Appendix A.4, we begin by computing $z_i = \|y_{s_i}\|_2^2$ for $i = 1, \ldots, D$, which can be done in $O(p)$ operations. Define $a_i = \|\beta^{i-1}_{g_i}\|_2^2$ and observe that $a_1 = z_1$ and that, for $i \geq 1$,

$$a_{i+1} = z_{i+1} + \|\beta^i_{g_i}\|_2^2 = z_{i+1} + (a_i^{1/2} - \lambda w_i)_+^2.$$

Thus, we can compute $a_1, \cdots, a_D$ in $O(D)$ operations. For $\ell = 1, \ldots, D$, we form $b_\ell = \prod_{i=\ell}^D \left(1 - \frac{\lambda w_i}{\sqrt{a_i}}\right)_+$ (which can be done in $O(D)$ operations) and observe that

$$\beta^D_{s_\ell} = b_\ell y_{s_\ell}.$$

This final scaling of the elements of $y$ takes $O(p)$. Thus, computing the GL proximal operator can be done in $O(p + D)$ operations.

### A.5.2 Modified GL Proximal Operator

When we introduced $\Omega^{d(\mathcal{D})}_{\text{mGL}}$ in (2.16) of Section **??**, we defined the penalty in the one parameter per node case. Following Bien et al. (2016), we now generalize the definition to the situation of multiple parameters per node in a directed path graph $\mathcal{D}$. For $\ell = 1, \ldots, D$, we let $g_\ell = s_{\ell:D}$. Let $w_{\ell,m} = \frac{\sqrt{|s_\ell|}}{m-\ell+1}$ where $1 \le \ell \le m \le D$ be the weight applied to $s_m$ in $g_\ell$. The modified GL penalty under a path graph can be written as

$$\Omega^{d(\mathcal{D})}_{\text{mGL}}(\beta; \{w_{\ell,m}\}) = \sum_{\ell=1}^{D} \sqrt{\sum_{m=\ell}^{D} w^2_{\ell,m} \|\beta_{s_m}\|_2^2}, \tag{A.18}$$

By Jenatton et al. (2011b)'s result, a single pass of BCD from $g_D$ to $g_1$ will solve the dual problem. Bien et al. (2016) proves the modified version of BCD in the context of covariance estimation, which itself is a special case of directed path graphs. By Theorem 2 of Bien et al. (2016), we have the algorithm stated in Algorithm 6:

**Algorithm 6** Solve Proximal Operator of Modified GL in (A.18)

$\beta^{D+1} \leftarrow y$
**for** $i = D, \ldots, 1$ **do**
  Solve $\lambda^2 = \sum_{m=i}^{D} \frac{w^2_{i,m}}{(w^2_{i,m}+\hat{\nu}^{(i)})^2} \left\|\beta^{i+1}_{s_m}\right\|_2^2$ for $\hat{\nu}^{(i)}$
  **for** $m = 1, \ldots, D$ **do**
    $\beta^i_{s_m} \leftarrow \frac{[\hat{\nu}^{(i)}]_+}{w^2_{i,m}+[\hat{\nu}^{(i)}]_+} \beta^{i+1}_{s_m}$
  **end for**
**end for**
**Output:** $\beta^1$

We can define $t \in \mathbb{R}^p$ such that for $m = 1, \ldots, D$,

$$(t_{s_m})_j = \begin{cases} \sum_{i=1}^{m} \frac{[\hat{\nu}^{(i)}]_+}{w^2_{i,m}+[\hat{\nu}^{(i)}]_+} & \text{if } j \in s_m \\ 0 & \text{otherwise .} \end{cases}$$

The solution $\hat{\beta}$ can be written as $\hat{\beta} = t * y$ where $*$ denotes elementwise multiplication. Provided all the $\{\hat{\nu}^{(i)}\}_{i=1,\ldots D}$ have been found, computing $t$ requires

$O\left(\sum_{m=1}^{D} m\right) = O(D^2)$ operations. Performing elementwise multiplication to get $\hat{\beta}$ can be done in $O(p)$ operations.

To find a root $\{\hat{\nu}^{(i)}\}_{i=1,\dots D}$, Bien et al. (2016) shows that $\hat{\nu}^{(i)} \leq 0$ when $\lambda^2 \geq \sum_{m=i}^{D} \|\beta_{s_m}^{i+1}\|_2^2 / w_{i,m}^2$. In that case, $\beta_{g_i}^i = 0$. If parameters corresponding to $\{g_D, \dots, g_{\hat{K}+1}\}$ are zeroed out, only the last $\hat{K}$ roots need to be numerically computed. We start by computing $z_i = \|y_{s_i}\|_2^2$ for $i = 1, \dots, D$, which can be done in $O(p)$ operations. Then do the following two steps:

1. Compute $z_i/|s_i|$ for $i = D, \dots, 1$. Let $i = \hat{K}$ be the first time $\lambda^2 < z_i/|s_i|$. The amount of operations is $O(D)$. At the end of this part, we have $\beta_{g_{\hat{K}+1}}^{\hat{K}+1} = 0$ if $\hat{K} < D$.

2. For $i \in \{\hat{K}, \dots, 1\}$, we need to find $\nu$ such that

$$f(\nu) = 1 - \frac{\lambda}{\sqrt{\sum_{m=i}^{D} \frac{w_{i,m}^2 \|\beta_{s_m}^{i+1}\|_2^2}{(w_{i,m}^2+\nu)^2}}} = 1 - \frac{\lambda}{\sqrt{\sum_{m=i}^{\hat{K}} \frac{w_{i,m}^2 \|\beta_{s_m}^{i+1}\|_2^2}{(w_{i,m}^2+\nu)^2}}} = 0,$$

   which can be solved using Newton's method. At each iteration of Newton's method, we need to compute

$$\frac{f(\nu)}{f'(\nu)} = \frac{\sum_{m=i}^{\hat{K}} \frac{w_{i,m}^2 \|\beta_{s_m}^{i+1}\|_2^2}{(w_{i,m}^2+\nu)^2} - \lambda^{-1} \left(\sum_{m=i}^{\hat{K}} \frac{w_{i,m}^2 \|\beta_{s_m}^{i+1}\|_2^2}{(w_{i,m}^2+\nu)^2}\right)^{1.5}}{\sum_{m=i}^{\hat{K}} \frac{w_{i,m}^2 \|\beta_{s_m}^{i+1}\|_2^2}{(w_{i,m}^2+\nu)^3}}.$$

   Evaluating $\|\beta_{s_m}^{i+1}\|_2^2$ can be done efficiently. For $i = \hat{K}, \dots, 1$ and $m = i, \dots, \hat{K}$, define $a^{(i,m)} = \|\beta_{s_m}^{i+1}\|_2^2$. It is obvious that $a^{(i,i)} = \|y_{s_i}\|_2^2 = z_i$ for $i = \hat{K}, \dots, 1$. For $m \geq i$, we have

$$a^{(i-1,m)} = \left\|\beta_{s_m}^{i}\right\|_2^2 = \left(\frac{[\hat{\nu}^{(i)}]_+}{w_{i,m}^2 + [\hat{\nu}^{(i)}]_+}\right)^2 a^{(i,m)}.$$

   Applying this update, we can compute all $\{a^{(i,m)}\}$ with $i \leq m$ in a total of $O\left(\sum_{m=1}^{\hat{K}} m\right) = O(\hat{K}^2)$ operations. At a fixed $i = \hat{K}, \dots, 1$, provided all the

needed $\{a^{(i,m)}\}$ are computed already, evaluating $f(v)/f'(v)$ requires $O(\hat{K}-i)$ per $v$ value. Newton's method is known for its quadratic convergence rate once the estimate gets "near" a root (Proposition 1.4.1 of Bertsekas 1999). Therefore, the number of significant digits double with each iteration when the estimate gets close to the root. For $n$-digit precision, Newton's method needs $O\left(\log(n)(\hat{K}-i)\right)$ operations if the initial point is good. Therefore, the total amount of computations for Step 2 is

$$O\left(\hat{K}^2 + \log(n)\sum_{i=1}^{\hat{K}}(\hat{K}-i)\right) = O\left(\log(n)\hat{K}^2\right) = O\left(D^2\log(n)\right).$$

Combing the above derivation, the proximal operator of modified GL can be computed in $O\left(p + D^2\log(n)\right)$ operations, where $n$ is the pre-determined number of digits of precision for Newton's method.

## A.6 Proof of Lemma 4

Recalling that $\mathcal{G}_1, \cdots, \mathcal{G}_L$ is a partition of $a(\mathcal{D})$, we can write Problem (2.8) as the following:

$$\begin{aligned}
&\min_{\beta\in\mathbb{R}^p}\left\{F(\beta) + \lambda\Omega_{\text{LOG}}^{a(\mathcal{D})}(\beta; w)\right\} \\
\Leftrightarrow &\min_{\{v^{(g)}\in\mathbb{R}^p\}_{g\in a(\mathcal{D})}}\left\{F\left(\sum_{\ell=1}^{L}\sum_{g\in\mathcal{G}_\ell} v^{(g)}\right) + \lambda\sum_{\ell=1}^{L}\sum_{g\in\mathcal{G}_\ell} w_g\left\|v^{(g)}\right\|_2 \quad \text{s.t.} \quad v_{g^c}^{(g)} = 0\ \forall g\in a(\mathcal{D})\right\} \\
\Leftrightarrow &\min_{\{v^{(g)}\in\mathbb{R}^p\}_{g\in a(\mathcal{D})}}\left\{F\left(\sum_{\ell=1}^{L}\beta^{(\ell)}\right) + \lambda\sum_{\ell=1}^{L}\sum_{g\in\mathcal{G}_\ell} w_g\left\|v^{(g)}\right\|_2 \quad \text{s.t.} \quad v_{g^c}^{(g)} = 0\ \forall g\in a(\mathcal{D}),\ \beta^{(\ell)} = \sum_{g\in\mathcal{G}_\ell} v^{(g)}\right\}.
\end{aligned} \tag{A.19}$$

Finally, by definition of the LOG penalty, we can write (A.19) as

$$\min_{\{\beta^{(\ell)}\in\mathbb{R}^p\}_{\ell=1}^{L}}\left\{F\left(\sum_{\ell=1}^{L}\beta^{(\ell)}\right) + \lambda\sum_{\ell=1}^{L}\Omega_{\text{LOG}}^{\mathcal{G}_\ell}(\beta^{(\ell)}; w_{\mathcal{P}_\ell}) \quad \text{s.t.} \quad \text{supp}(\beta^{(\ell)}) \subset \bigcup_{g\in\mathcal{G}_\ell} g\right\},$$

where $w_{\mathcal{P}_\ell} = \{w_g : g \in \mathcal{G}'_\ell\}$.

## A.7 Simple Algorithm for Path Decomposition of DAG

Algorithm 7 presents a simple greedy algorithm for decomposing $\mathcal{D}$ into paths.

**Algorithm 7** Path Decomposition of a DAG $\mathcal{D}$

**Input:** $\mathcal{D}$

$\mathcal{M} \leftarrow \emptyset$ and $L \leftarrow 1$
Form set of "root nodes" $R = \{s_i : \text{ancestors}(\mathcal{D}; s_i) = \{s_i\}\}$.
**for** $s_i \in R$ **do**
    **while** descendants$(\mathcal{D}; s_i) \not\subseteq \mathcal{M}$ **do**
        Choose the path $\mathcal{P}$ from $s_i$ for which $|\mathcal{P} \setminus \mathcal{M}|$ is largest.
        Define $\mathcal{P}_\ell \leftarrow \mathcal{P} \setminus \mathcal{M}$
        $\mathcal{M} \leftarrow \mathcal{M} \cup \mathcal{P}_\ell$.
        $L \leftarrow L + 1$
    **end while**
**end for**

**Output:** $\mathcal{P}_1, \cdots, \mathcal{P}_L$.

## A.8 Proof of Lemma 5

By Lemma 4, Problem (2.8) with $F(\beta) = \frac{1}{2}\|y - \boldsymbol{X}\beta\|_2^2$ can be written in terms of $\{\beta^{(\ell)}\}_{\ell=1}^L$ subject to $\beta = \sum_{\ell=1}^L \beta^{(\ell)}$:

$$\min_{\{\beta^{(\ell)} \in \mathbb{R}^p\}_{\ell=1}^L} \frac{1}{2} \left\| y - \mathbf{X} \sum_{\ell=1}^L \beta^{(\ell)} \right\|_2^2 + \lambda \sum_{\ell=1}^L \Omega_{\text{LOG}}^{\mathcal{G}_\ell}(\beta^{(\ell)}; w_{\mathcal{P}_\ell})$$
$$\text{s.t.} \quad \text{supp}(\beta^{(\ell)}) \subseteq g^{(\ell)} \quad \forall \ell = 1, \ldots, L. \tag{A.20}$$

Then (2.19) follows by substituting $\{\beta^{(\ell)}\}$ with $\{\gamma^{(\ell)}\}$ in the squared loss of (A.20). The augmented Lagrangian subject to $\text{supp}(\beta^{(\ell)}) \subseteq g^{(\ell)}$ and $\text{supp}(\gamma^{(\ell)}) \subseteq g^{(\ell)}$ $\forall \ell$ is

$$
\begin{aligned}
& L(\{\beta^{(\ell)}\}, \{\gamma^{(\ell)}\}, \{u^{(\ell)}\}) \\
=& \frac{1}{2}\left\| y - \mathbf{X}\sum_{\ell=1}^{L}\gamma^{(\ell)} \right\|_2^2 + \lambda \sum_{\ell=1}^{L} \Omega_{\text{LOG}}^{\mathcal{G}_\ell}(\beta^{(\ell)}; w_{\mathcal{P}_\ell}) + \left\langle \begin{pmatrix} u^{(1)} \\ \vdots \\ u^{(L)} \end{pmatrix}, \begin{pmatrix} \beta^{(1)} - \gamma^{(1)} \\ \vdots \\ \beta^{(L)} - \gamma^{(L)} \end{pmatrix} \right\rangle + \frac{\rho}{2}\left\| \begin{pmatrix} \beta^{(1)} - \gamma^{(1)} \\ \vdots \\ \beta^{(L)} - \gamma^{(L)} \end{pmatrix} \right\|_2^2 \\
=& \frac{1}{2}\left\| y - \mathbf{X}\sum_{\ell=1}^{L}\gamma^{(\ell)} \right\|_2^2 + \lambda \sum_{\ell=1}^{L} \Omega_{\text{LOG}}^{\mathcal{G}_\ell}(\beta^{(\ell)}; w_{\mathcal{P}_\ell}) + \frac{\rho}{2}\sum_{\ell=1}^{L}\left\| \beta^{(\ell)} - \gamma^{(\ell)} + \frac{1}{\rho}u^{(\ell)} \right\|_2^2 - \frac{1}{2\rho}\sum_{\ell=1}^{L}\left\| u^{(\ell)} \right\|_2^2 .
\end{aligned}
$$

Alternating Direction Method of Multipliers (ADMM) iteratively updates $\{\gamma^{(\ell)}\}$ and $\{\beta^{(\ell)}\}$ by optimizing the corresponding part in the augmented Lagrangian.

**Step 1**: Optimize over $\{\gamma^{(\ell)}\}$. For $\ell = 1, \ldots, L$,

$$
\begin{aligned}
\hat{\gamma}^{(\ell)} = \underset{\gamma^{(\ell)} \in \mathbb{R}^p}{\arg\min}\ & \frac{1}{2}\left\| y - \mathbf{X}\sum_{\ell'=1}^{L}\gamma^{(\ell')} \right\|_2^2 + \frac{\rho}{2}\left\| \hat{\beta}^{(\ell)} - \gamma^{(\ell)} + \frac{1}{\rho}\hat{u}^{(\ell)} \right\|_2^2 \\
\text{s.t.} \quad & \text{supp}(\gamma^{(\ell)}) \subseteq g^{(\ell)}.
\end{aligned}
$$

Solving the gradient with respect to $\gamma^{(\ell)}_{|g^{(\ell)}}$ equal to zero yields:

$$
\mathbf{X}^T_{|g^{(\ell)}}\left( \mathbf{X}\sum_{\ell'}\gamma^{(\ell')} - y \right) + \rho\left( \gamma^{(\ell)}_{|g^{(\ell)}} - \hat{\beta}^{(\ell)}_{|g^{(\ell)}} - \frac{1}{\rho}\hat{u}^{(\ell)}_{|g^{(\ell)}} \right) = 0.
$$

It follows that

$$
\begin{aligned}
\gamma^{(\ell)}_{|g^{(\ell)}} &= \hat{\beta}^{(\ell)}_{|g^{(\ell)}} + \frac{1}{\rho}\hat{u}^{(\ell)}_{|g^{(\ell)}} + \frac{1}{\rho}\mathbf{X}^T_{|g^{(\ell)}}\left( y - \mathbf{X}\sum_{\ell'}\gamma^{(\ell')} \right) \\
&= \hat{\beta}^{(\ell)}_{|g^{(\ell)}} + \frac{1}{\rho}\hat{u}^{(\ell)}_{|g^{(\ell)}} + \frac{1}{\rho}\mathbf{X}^T_{|g^{(\ell)}}\left( y - \sum_{\ell'}\mathbf{X}_{|g^{(\ell')}}\gamma^{(\ell')}_{|g^{(\ell')}} \right). \qquad \text{(A.21)}
\end{aligned}
$$

Left-multiplying both sides of (A.21) by $\mathbf{X}_{|g^{(\ell)}}$ yields

$$
\mathbf{X}_{|g^{(\ell)}}\gamma^{(\ell)}_{|g^{(\ell)}} = \mathbf{X}_{|g^{(\ell)}}\left( \hat{\beta}^{(\ell)}_{|g^{(\ell)}} + \frac{1}{\rho}\hat{u}^{(\ell)}_{|g^{(\ell)}} \right) + \frac{1}{\rho}\mathbf{X}_{|g^{(\ell)}}\mathbf{X}^T_{|g^{(\ell)}}\left( y - \sum_{\ell'}\mathbf{X}_{|g^{(\ell')}}\gamma^{(\ell')}_{|g^{(\ell')}} \right). \qquad \text{(A.22)}
$$

Summing up (A.22) over all $\ell'$s yields

$$\begin{aligned}
&\sum_{\ell} \mathbf{X}_{|g^{(\ell)}} \gamma^{(\ell)}_{|g^{(\ell)}} = \sum_{\ell} \left[ \mathbf{X}_{|g^{(\ell)}} \left( \hat{\beta}^{(\ell)}_{|g^{(\ell)}} + \frac{1}{\rho} \hat{u}^{(\ell)}_{|g^{(\ell)}} \right) + \frac{1}{\rho} \mathbf{X}_{|g^{(\ell)}} \mathbf{X}^T_{|g^{(\ell)}} y \right] - \frac{1}{\rho} \sum_{\ell} \mathbf{X}_{|g^{(\ell)}} \mathbf{X}^T_{|g^{(\ell)}} \sum_{\ell'} \mathbf{X}_{|g^{(\ell')}} \gamma^{(\ell')}_{|g^{(\ell')}} \\
&\Rightarrow \left( I + \frac{1}{\rho} \sum_{\ell} \mathbf{X}_{|g^{(\ell)}} \mathbf{X}^T_{|g^{(\ell)}} \right) \sum_{\ell} \mathbf{X}_{|g^{(\ell)}} \gamma^{(\ell)}_{|g^{(\ell)}} = \sum_{\ell} \left[ \mathbf{X}_{|g^{(\ell)}} \left( \hat{\beta}^{(\ell)}_{|g^{(\ell)}} + \frac{1}{\rho} \hat{u}^{(\ell)}_{|g^{(\ell)}} \right) + \frac{1}{\rho} \mathbf{X}_{|g^{(\ell)}} \mathbf{X}^T_{|g^{(\ell)}} y \right] \\
&\Rightarrow \sum_{\ell} \mathbf{X}_{|g^{(\ell)}} \gamma^{(\ell)}_{|g^{(\ell)}} = \left( I + \frac{1}{\rho} \sum_{\ell} \mathbf{X}_{|g^{(\ell)}} \mathbf{X}^T_{|g^{(\ell)}} \right)^{-1} \sum_{\ell} \left[ \mathbf{X}_{|g^{(\ell)}} \left( \hat{\beta}^{(\ell)}_{|g^{(\ell)}} + \frac{1}{\rho} \hat{u}^{(\ell)}_{|g^{(\ell)}} \right) + \frac{1}{\rho} \mathbf{X}_{|g^{(\ell)}} \mathbf{X}^T_{|g^{(\ell)}} y \right].
\end{aligned} \tag{A.23}$$

Substituting (A.23) into (A.21) yields

$$\hat{\gamma}^{(\ell)}_{|g^{(\ell)}} = \hat{\beta}^{(\ell)}_{|g^{(\ell)}} + \frac{1}{\rho} \hat{u}^{(\ell)}_{|g^{(\ell)}} + \frac{1}{\rho} \mathbf{X}^T_{|g^{(\ell)}} (y - \Delta),$$

where $\Delta := \sum_{\ell} \mathbf{X}_{|g^{(\ell)}} \gamma^{(\ell)}_{|g^{(\ell)}}$ in (A.23).

**Step 2**: Optimize over $\{\beta^{(\ell)}\}$. For $\ell = 1, \ldots, L$,

$$\begin{aligned}
\hat{\beta}^{(\ell)} &= \underset{\beta^{(\ell)} \in \mathbb{R}^p}{\arg\min} \sum_{\ell=1}^{L} \left\{ \frac{1}{2} \left\| \beta^{(\ell)} - \left( \hat{\gamma}^{(\ell)} - \frac{1}{\rho} \hat{u}^{(\ell)} \right) \right\|_2^2 + \frac{\lambda}{\rho} \Omega^{\mathcal{G}_\ell}_{\mathrm{LOG}}(\beta^{(\ell)}; w_{\mathcal{P}_\ell}) \right\} \\
&\qquad \text{s.t.} \quad \mathrm{supp}(\beta^{(\ell)}) \subseteq g^{(\ell)} \\
\hat{\beta}^{(\ell)}_{|g^{(\ell)}} &= \mathrm{Prox}^{\mathcal{G}_\ell}_{\mathrm{LOG}} \left( \left( \hat{\gamma}^{(\ell)}_{|g^{(\ell)}} - \frac{1}{\rho} \hat{u}^{(\ell)}_{|g^{(\ell)}} \right); \frac{\lambda}{\rho}, w_{\mathcal{P}_\ell} \right).
\end{aligned}$$

All the $\hat{\beta}^{(\ell)}_{|g^{(\ell)}}$'s can be efficiently updated using path-based BCD in Algorithm 4.

**Step 3**: $\hat{u}^{(\ell)} \leftarrow \hat{u}^{(\ell)} + \rho(\hat{\gamma}^{(\ell)} - \hat{\beta}^{(\ell)})$ for $\ell = 1, \ldots, L$.

## A.9 Proof of Theorem 1

If $K = p - 1$, then $\hat{K} \leq K$.

If $K < p-1$, let $\bar{K}$ be the largest knot such that $\bar{K} \leq K$. Then $\hat{K} \geq \bar{K}$. We will show that $\forall k > K$

$$\frac{\left\|\mathbf{S}_{s_{(\bar{K}+1):k}}\right\|_F^2}{\left|s_{(\bar{K}+1):k}\right|} \leq \lambda^2. \tag{A.24}$$

through the following two cases.

**Case 1**: If $\bar{K} = K$, then $\forall k > K$, we have

$$\frac{\left\|\mathbf{S}_{s_{(\bar{K}+1):k}}\right\|_F^2}{\left|s_{(\bar{K}+1):k}\right|} = \frac{\left\|\mathbf{S}_{s_{(\bar{K}+1):k}} - \mathbf{\Sigma}^*_{s_{(\bar{K}+1):k}}\right\|_F^2}{\left|s_{(\bar{K}+1):k}\right|} \leq \max_{ij} |\mathbf{S}_{ij} - \mathbf{\Sigma}^*_{ij}|^2 \leq \lambda^2. \tag{A.25}$$

**Case 2**: If $\bar{K} < K$, then $\forall k > K$, we have

$$\left\|\mathbf{S}_{s_{(\bar{K}+1):k}}\right\|_F^2 = \left\|\mathbf{S}_{s_{(\bar{K}+1):K}}\right\|_F^2 + \left\|\mathbf{S}_{s_{(K+1):k}} - \mathbf{\Sigma}^*_{s_{(K+1):k}}\right\|_F^2. \tag{A.26}$$

Since $\bar{K}$ is the largest knot before or at $K$, by Algorithm 8 we have $\forall i = \bar{K}+1, \ldots, K$ either **(a)** or **(b)** is true.

**(a)** $\left\|\mathbf{S}_{s_{(\bar{K}+1):i}}\right\|_F \leq \lambda \left|s_{(\bar{K}+1):i}\right|^{1/2}$

**(b)** $\exists \bar{k} > i$ s.t. $\left\|\mathbf{S}_{s_{(\bar{K}+1):i}}\right\|_F \leq \left\|\mathbf{S}_{s_{(\bar{K}+1):\bar{k}}}\right\|_F \frac{\left|s_{(\bar{K}+1):i}\right|^{1/2}}{\left|s_{(\bar{K}+1):\bar{k}}\right|^{1/2}}$

If (**a**) holds for $i = K$, then (A.26) becomes

$$\begin{aligned}\left\|\mathbf{S}_{s_{(\bar{K}+1):k}}\right\|_F^2 &\leq \lambda^2 \left|s_{(\bar{K}+1):K}\right| + \left\|\mathbf{S}_{s_{(K+1):k}} - \mathbf{\Sigma}^*_{s_{(K+1):k}}\right\|_F^2 \\ &\leq \lambda^2 \left|s_{(\bar{K}+1):K}\right| + \lambda^2 \left|s_{(K+1):k}\right| = \lambda^2 \left|s_{(\bar{K}+1):k}\right|.\end{aligned}$$

If (**b**) holds for $i = K$, then $\exists \bar{k} > K$ such that

$$\left\|\mathbf{S}_{s_{(\bar{K}+1):K}}\right\|_F^2 \leq \left\|\mathbf{S}_{s_{(\bar{K}+1):\bar{k}}}\right\|_F^2 \frac{\left|s_{(\bar{K}+1):K}\right|}{\left|s_{(\bar{K}+1):\bar{k}}\right|} = \left(\left\|\mathbf{S}_{s_{(\bar{K}+1):K}}\right\|_F^2 + \left\|\mathbf{S}_{s_{(K+1):\bar{k}}}\right\|_F^2\right) \frac{\left|s_{(\bar{K}+1):K}\right|}{\left|s_{(\bar{K}+1):\bar{k}}\right|}$$

Let $\alpha = \frac{\left|s_{(\bar{K}+1):K}\right|}{\left|s_{(\bar{K}+1):\bar{k}}\right|}$. Then,

$$\left\|\mathbf{S}_{s_{(\bar{K}+1):K}}\right\|_F^2 (1-\alpha) \quad \leq \quad \left\|(\mathbf{S} - \mathbf{\Sigma}^*)_{(K+1):\bar{k}}\right\|_F^2 \alpha$$

$$\Rightarrow \quad \left\|\mathbf{S}_{s_{(\bar{K}+1):K}}\right\|_F^2 \quad \leq \quad \left(\frac{\alpha}{1-\alpha}\right)\lambda^2\left|s_{(K+1):\bar{k}}\right|. \tag{A.27}$$

Let $a = \left|s_{(\bar{K}+1):K}\right|$ and $b = \left|s_{(K+1):\bar{k}}\right|$. Then $\alpha = \frac{a}{a+b}$. It can be derived that $\left(\frac{\alpha}{1-\alpha}\right)b = a$. Therefore,

$$\left(\frac{\alpha}{1-\alpha}\right)b \leq a \quad \Rightarrow \quad \left(\frac{\alpha}{1-\alpha}\right)\left|s_{(K+1):\bar{k}}\right| \leq \left|s_{(\bar{K}+1):K}\right|. \tag{A.28}$$

Combining (A.27) and (A.28) yields

$$\left\|\mathbf{S}_{s_{(\bar{K}+1):K}}\right\|_F^2 \leq \left(\frac{\alpha}{1-\alpha}\right)\lambda^2\left|s_{(K+1):\bar{k}}\right| \leq \lambda^2\left|s_{(\bar{K}+1):K}\right|. \tag{A.29}$$

Considering $\left\|\mathbf{S}_{s_{(K+1):k}}\right\|_F^2 = \left\|\mathbf{S}_{s_{(K+1):k}} - \mathbf{\Sigma}^*_{s_{(K+1):k}}\right\|_F^2 \leq \lambda^2\left|s_{(K+1):k}\right|$ and (A.29), we have

$$\left\|\mathbf{S}_{s_{(\bar{K}+1):k}}\right\|_F^2 \leq \lambda^2\left|s_{(\bar{K}+1):k}\right|.$$

In both Case 1 and Case 2, we have $\frac{\left\|\mathbf{S}_{s_{(\bar{K}+1):k}}\right\|_F^2}{|s_{(\bar{K}+1):k}|} \leq \lambda^2$. By Algorithm 8, $\bar{K}$ is the last knot in both cases. Hence, $\hat{K} = \bar{K} \leq K$.

## A.10 Proof of Theorem 2

Let $\tilde{K}$ be the largest knot such that $\tilde{K} < K$. Being on the set $\mathcal{A}_x$ implies that for any $k > \tilde{K}$,

$$\begin{aligned}\left\|\mathbf{S}_{s_{(\tilde{K}+1):k}}\right\|_F &\geq \left\|\mathbf{\Sigma}^*_{s_{(\tilde{K}+1):k}}\right\|_F - \left\|\mathbf{S}_{s_{(\tilde{K}+1):k}} - \mathbf{\Sigma}^*_{s_{(\tilde{K}+1):k}}\right\|_F \\ &\geq \left\|\mathbf{\Sigma}^*_{s_{(\tilde{K}+1):k}}\right\|_F - \lambda\sqrt{|s_{(\tilde{K}+1):k}|}.\end{aligned} \tag{A.30}$$

From (A.30), we have

$$\max_{k\geq K}\left\{\frac{\left\|\mathbf{S}_{s_{(\tilde{K}+1):k}}\right\|_F}{\left|s_{(\tilde{K}+1):k}\right|^{\frac{1}{2}}}\right\} \geq \max_{k\geq K}\left\{\frac{\left\|\mathbf{\Sigma}^*_{s_{(\tilde{K}+1):k}}\right\|_F}{\left|s_{(\tilde{K}+1):k}\right|^{\frac{1}{2}}}\right\} - \lambda \geq \frac{\left\|\mathbf{\Sigma}^*_{s_{(\tilde{K}+1):K}}\right\|_F}{\left|s_{(\tilde{K}+1):K}\right|^{\frac{1}{2}}} - \lambda > 2\lambda - \lambda = \lambda. \tag{A.31}$$

where the last equality holds by Assumption (2.21), given $\tilde{K} + 1 \leq K$. Equivalently, $\exists k \geq K$ such that

$$\frac{\left\|\mathbf{S}_{s_{(\tilde{K}+1):k}}\right\|_F^2}{|s_{(\tilde{K}+1):k}|} > \lambda^2. \tag{A.32}$$

There exists a knot $k \geq K$ when applying Algorithm 8 to solve the problem. Hence, $\hat{K} \geq K$.

## A.11 Proof of Theorem 3

We can rewrite Problem (2.20) in terms of the latent variables $\{\mathbf{V}^{(k)}\}_{k=1}^{p-1}$:

$$\{\hat{\mathbf{V}}^{(k)}\}_{k=1}^{p-1} = \underset{\mathbf{V}^{(1)},\cdots,\mathbf{V}^{(p-1)}\in\mathbb{R}^{p\times p}}{\arg\min} \left\{ \frac{1}{2} \left\| \sum_{k=1}^{p-1} \mathbf{V}^{(k)} - \mathbf{S}^{-} \right\|_F^2 + \lambda \sum_{k=1}^{p-1} w_k \left\| \mathbf{V}^{(k)} \right\|_F \text{ s.t. } \operatorname{supp}(\mathbf{V}^{(k)}) \subseteq s_{1:k} \right\} \tag{A.33}$$

so that $\hat{\mathbf{\Sigma}}^{\text{LOG}-} = \sum_{k=1}^{p-1} \hat{\mathbf{V}}^{(k)}$. In addition, $\hat{\mathbf{\Sigma}}^{\text{LOG}}_{s_0} = \mathbf{S}_{s_0}$ because the LOG penalty does not apply to the diagonal elements. Taking subgradient of the objective function in (A.33) with respect to $\mathbf{V}^{(K)}$ where $K$ is the bandwidth of $\mathbf{\Sigma}^*$ yields:

$$0 \in \left( \sum_{k=1}^{p-1} \hat{\mathbf{V}}^{(k)} - \mathbf{S}^{-} \right)_{s_{1:K}} + \lambda w_K \partial \left\| \mathbf{V}^{(K)} \right\|_F . \tag{A.34}$$

When $\mathbf{V}^{(K)} \neq 0$,

$$\partial \left\| \mathbf{V}^{(K)} \right\|_F = \frac{\mathbf{V}^{(K)}}{\left\| \mathbf{V}^{(K)} \right\|_F}. \tag{A.35}$$

When $\mathbf{V}^{(K)} = 0$,

$$\begin{aligned} \partial \left\| \mathbf{V}^{(K)} \right\|_F &= \left\{ \mathbf{Z} \in \mathbb{R}^{p\times p} : \|\mathbf{U}\|_F \geq \left\| \mathbf{V}^{(K)} \right\|_F + \langle \mathbf{Z}, \mathbf{U} - \mathbf{V}^{(k)} \rangle \ \forall \mathbf{U} \in \mathbb{R}^{p\times p} \right\} \\ &= \left\{ \mathbf{Z} \in \mathbb{R}^{p\times p} : \|\mathbf{U}\|_F \geq \langle \mathbf{Z}, \mathbf{U} \rangle \ \forall \mathbf{U} \in \mathbb{R}^{p\times p} \right\} \\ &= \left\{ \mathbf{Z} \in \mathbb{R}^{p\times p} : \|\mathbf{Z}\|_F \leq 1 \right\}. \end{aligned} \tag{A.36}$$

Combining (A.34), (A.35) and (A.36) we have

$$\begin{aligned} \left\| \left( \sum_{k=1}^{p-1} \hat{\mathbf{V}}^{(k)} - \mathbf{S}^{-} \right)_{s_{1:K}} \right\|_F \leq \lambda w_K \quad &\Leftrightarrow \quad \left\| \left( \hat{\mathbf{\Sigma}}^{\text{LOG}-} - \mathbf{S}^{-} \right)_{s_{1:K}} \right\|_F \leq \lambda w_K \\ &\Leftrightarrow \quad \left\| \left( \hat{\mathbf{\Sigma}}^{\text{LOG}} - \mathbf{S} \right)_{s_{1:K}} \right\|_F \leq \lambda w_K. \end{aligned} \tag{A.37}$$

Furthermore, on $\mathcal{A}_x$ we have

$$\lambda^2 \geq \max_{i=j} \left|\mathbf{S}_{ij} - \mathbf{\Sigma}^*_{ij}\right|^2 \geq \frac{1}{p} \left\|\mathbf{S}_{s_0} - \mathbf{\Sigma}^*_{s_0}\right\|_F^2, \tag{A.38}$$

$$\lambda \geq \max_{i,j} \left|\mathbf{S}_{ij} - \mathbf{\Sigma}^*_{ij}\right| \geq \frac{1}{\sqrt{|s_{1:K}|}} \left\|(\mathbf{S} - \mathbf{\Sigma}^*)_{s_{1:K}}\right\|_F. \tag{A.39}$$

Using triangle inequality, (A.37) and (A.39) we have

$$\begin{aligned} \left\|(\hat{\mathbf{\Sigma}}^{\mathrm{LOG}} - \mathbf{\Sigma}^*)_{s_{1:K}}\right\|_F &\leq \left\|(\hat{\mathbf{\Sigma}}^{\mathrm{LOG}} - \mathbf{S})_{s_{1:K}}\right\|_F + \left\|(\mathbf{S} - \mathbf{\Sigma}^*)_{s_{1:K}}\right\|_F \\ &\leq \lambda w_K + \lambda \sqrt{|s_{1:K}|} = 2\lambda \sqrt{|s_{1:K}|}. \end{aligned} \tag{A.40}$$

Using (A.38) and (A.40) we have

$$\begin{aligned} \left\|\hat{\mathbf{\Sigma}}^{\mathrm{LOG}} - \mathbf{\Sigma}^*\right\|_F^2 &= \left\|\left(\hat{\mathbf{\Sigma}}^{\mathrm{LOG}} - \mathbf{\Sigma}^*\right)_{s_{1:K}}\right\|_F^2 + \left\|\hat{\mathbf{\Sigma}}^{\mathrm{LOG}}_{s_0} - \mathbf{\Sigma}^*_{s_0}\right\|_F^2 \\ &= \left\|\left(\hat{\mathbf{\Sigma}}^{\mathrm{LOG}} - \mathbf{\Sigma}^*\right)_{s_{1:K}}\right\|_F^2 + \left\|\mathbf{S}_{s_0} - \mathbf{\Sigma}^*_{s_0}\right\|_F^2 \\ &\leq 4\lambda^2 |s_{1:K}| + \lambda^2 p \\ &\leq \frac{4x^2 pK \log p}{n} + \frac{x^2 p \log p}{n}. \end{aligned} \tag{A.41}$$

By Theorem 1, $\hat{K} \leq K$ with high probability when $\lambda \geq x\sqrt{\log p/n}$. Therefore, the equality in (A.41) holds with high probability. Hence, $\left\|\hat{\mathbf{\Sigma}}^{\mathrm{LOG}} - \mathbf{\Sigma}^*\right\|_F^2 \lesssim pK \log p/n$.

## A.12 Algorithm 8 for Solving Problem (2.20)

## A.13 PSD Probability (Figure A.1) and Minimum Eigenvalues (Figure A.2) of the Three Covariance Estimators

**Algorithm 8** Solve for $\hat{\mathbf{\Sigma}}^{\text{LOG}}$ defined by Problem (2.20)

**Input:** $\lambda \geq 0$, $\mathbf{S} \in \mathbb{R}^{p \times p}$ and $a(\mathcal{D})$.

$\mathbf{\Sigma} \leftarrow \mathbf{S}_{s_0}$
$k \leftarrow 0$
**while** $k < p - 1$ **do**
  $K \leftarrow \arg\max_{j:\, j>k} f(j,k)$ ▷ $f(j,k) = \dfrac{\left\|\mathbf{S}_{s_{(k+1):j}}\right\|_F}{\sqrt{\left|s_{(k+1):j}\right|}}$ for $0 \leq k < j \leq p-1$
  **if** $f(K,k) \leq \lambda$ **then**
    **break**
  **end if**
  $\mathbf{\Sigma}_{s_{(k+1):K}} \leftarrow S_G\left(\mathbf{S}_{s_{(k+1):K}}, \lambda\sqrt{\left|s_{(k+1):K}\right|}\right)$
  $k \leftarrow K$
**end while**

**Output:** $\mathbf{\Sigma}$

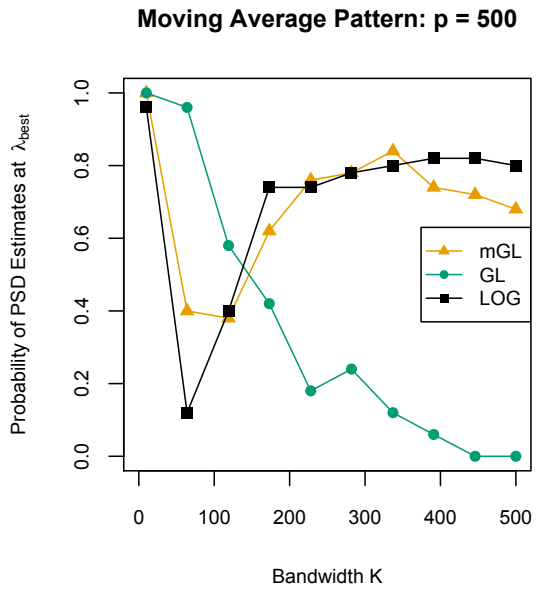


Figure A.1: For the three estimators ($\hat{\mathbf{\Sigma}}^{\text{mGL}}, \hat{\mathbf{\Sigma}}^{\text{GL}}, \hat{\mathbf{\Sigma}}^{\text{LOG}}$) in moving-average pattern, probability of their estimates being PSD at $\lambda_{best}$.

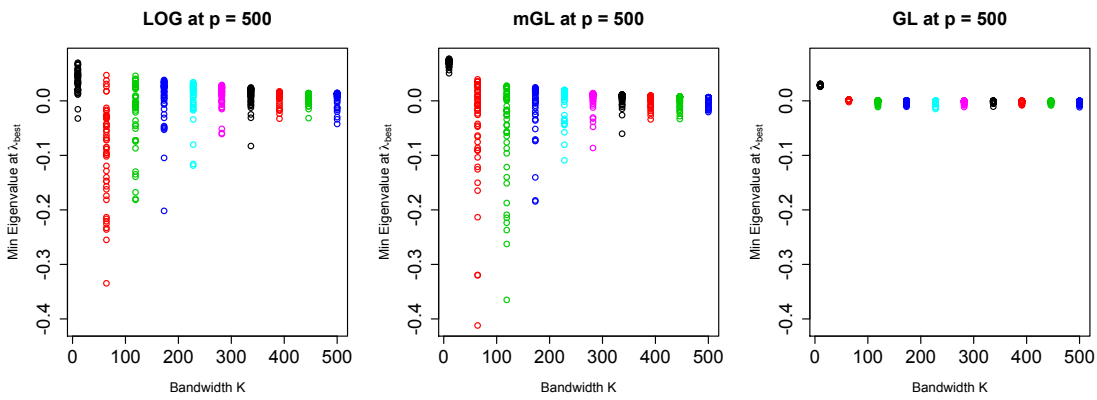


Figure A.2: For the three estimators ($\hat{\mathbf{\Sigma}}^{\text{LOG}}, \hat{\mathbf{\Sigma}}^{\text{mGL}}, \hat{\mathbf{\Sigma}}^{\text{GL}}$) in moving-average pattern, minimum eigenvalues of 50 samples at $\lambda_{best}$.

APPENDIX B

**APPENDIX FOR CHAPTER 3**

## B.1 Failure of OLS in the Presence of A Rare Feature

**Theorem 7.** *Consider the linear model* (3.1) *with* $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ *having full column rank. Further suppose that* $\boldsymbol{X}_j$ *is a binary vector having k nonzeros. It follows that*

$$\mathbb{P}\left(\left|\hat{\beta}_j^{OLS}(n) - \beta_j^*\right| > \eta\right) \geq 2\Phi\left(-\eta k^{1/2}/\sigma\right) \quad \textit{for any } \eta > 0, \tag{B.1}$$

*where* $\Phi(\cdot)$ *is the cumulative distribution function of a standard normal variable.*

*Proof.* The distribution of the OLS estimator is $\hat{\beta}_j^{\mathrm{OLS}}(n) \sim N(\beta_j^*, \sigma^2[(\boldsymbol{X}^T\boldsymbol{X})^{-1}]_{jj})$. By applying blockwise inversion (see, e.g., Bernstein 2009), with the $j$th row/column of $\boldsymbol{X}^T\boldsymbol{X}$ in its own "block", we get

$$\begin{aligned}
[(\boldsymbol{X}^T\boldsymbol{X})^{-1}]_{jj} &= [\boldsymbol{X}_j^T\boldsymbol{X}_j - \boldsymbol{X}_j^T\boldsymbol{X}_{-j}(\boldsymbol{X}_{-j}^T\boldsymbol{X}_{-j})^{-1}\boldsymbol{X}_{-j}^T\boldsymbol{X}_j]^{-1} \\
&= [\|\boldsymbol{X}_j\|^2 - \|(\boldsymbol{X}_{-j}^T\boldsymbol{X}_{-j})^{-1/2}\boldsymbol{X}_{-j}^T\boldsymbol{X}_j\|^2]^{-1} \\
&\geq \|\boldsymbol{X}_j\|^{-2} = k^{-1}.
\end{aligned}$$

Thus,

$$\mathbb{P}\left(\left|\hat{\beta}_j^{\mathrm{OLS}}(n) - \beta_j^*\right| > \eta\right) = 2\Phi\left(-\frac{\eta}{\sigma\sqrt{[(\boldsymbol{X}^T\boldsymbol{X})^{-1}]_{jj}}}\right) \geq 2\Phi\left(-\eta k^{1/2}/\sigma\right)$$

where $\Phi(\cdot)$ is the distribution function of a standard normal variable. □

## B.2 Proof of Theorem 5

In the setting of Theorem 5, we have $\boldsymbol{X} = \boldsymbol{I}_n \in \mathbb{R}^{n\times n}$ for $\boldsymbol{\beta}^*$ and $\widetilde{\boldsymbol{X}} = \boldsymbol{I}_k \otimes \mathbf{1}_{n/k} \in \mathbb{R}^{n\times k}$ for $\tilde{\boldsymbol{\beta}}^*$. Clearly $\boldsymbol{X}\boldsymbol{\beta}^* = \widetilde{\boldsymbol{X}}\tilde{\boldsymbol{\beta}}^*$. The two estimators, oracle lasso on the aggregated

data ($\widetilde{\boldsymbol{X}}$) and lasso on the original data ($\boldsymbol{X}$), are defined below.

- Oracle lasso estimator $\hat{\boldsymbol{\beta}}_\lambda^{oracle} = \check{\boldsymbol{\beta}}_\lambda^{oracle} \otimes 1_{n/k}$ where $\check{\boldsymbol{\beta}}_\lambda^{oracle}$ is the unique solution to
$$\min_{\tilde{\boldsymbol{\beta}} \in \mathbb{R}^k} \frac{1}{2n} \|\boldsymbol{y} - \widetilde{\boldsymbol{X}}\tilde{\boldsymbol{\beta}}\|_2^2 + \lambda \|\tilde{\boldsymbol{\beta}}\|_1.$$
- Lasso estimator $\hat{\boldsymbol{\beta}}_\lambda^{lasso}$ is defined in (3.2).

**Proposition 3** (Support recovery of oracle lasso)**.** *Suppose* $\min_{i=1,\dots,k-1} \left|\tilde{\beta}_i^*\right| > \sigma\sqrt{\frac{4k\log(k^2n)}{n}}$. *With* $\lambda = \sigma\sqrt{\frac{\log(k^2n)}{kn}}$, *the oracle lasso recovers the correct signed support successfully:*
$$\lim_{n\to\infty} \mathbb{P}\left(\mathbb{S}_\pm(\hat{\boldsymbol{\beta}}_\lambda^{oracle}) = \mathbb{S}_\pm(\boldsymbol{\beta}^*)\right) = 1.$$

*Proof.* Since $\widetilde{\boldsymbol{X}}^T\widetilde{\boldsymbol{X}} = \frac{n}{k}\boldsymbol{I}_k$, the scaled matrix $\sqrt{\frac{k}{n}}\widetilde{\boldsymbol{X}}$ is orthogonal. Orthogonality implies that
$$\check{\boldsymbol{\beta}}_\lambda^{oracle} = S\left(\left(\sqrt{\frac{k}{n}}\widetilde{\boldsymbol{X}}^T\right)\left(\sqrt{\frac{k}{n}}\boldsymbol{y}\right), \lambda k\right) = S\left(\frac{k}{n}\widetilde{\boldsymbol{X}}^T\boldsymbol{y}, \lambda k\right) \tag{B.2}$$
where $\frac{k}{n}\widetilde{\boldsymbol{X}}^T\boldsymbol{y} = \frac{k}{n}\widetilde{\boldsymbol{X}}^T\widetilde{\boldsymbol{X}}\tilde{\boldsymbol{\beta}}^* + \frac{k}{n}\widetilde{\boldsymbol{X}}^T\boldsymbol{\varepsilon} = \tilde{\boldsymbol{\beta}}^* + \frac{k}{n}\widetilde{\boldsymbol{X}}^T\boldsymbol{\varepsilon} \sim N_k(\tilde{\boldsymbol{\beta}}^*, \frac{k\sigma^2}{n}\boldsymbol{I}_k)$. By the Chernoff bound for normal variables, for any $t > 0$,
$$\mathbb{P}\left(\left|\frac{k}{n}(\widetilde{\boldsymbol{X}}_j)^T\boldsymbol{y} - \tilde{\beta}_j^*\right| > t\right) \le 2\exp\left(-\frac{t^2}{2k\sigma^2/n}\right) \quad \text{for } j = 1, \dots, k.$$
Choosing $t = \sigma\sqrt{\frac{k\log(k^2n)}{n}}$ and applying a union bound yields
$$\mathbb{P}\left(\left\|\frac{k}{n}\widetilde{\boldsymbol{X}}^T\boldsymbol{y} - \tilde{\boldsymbol{\beta}}^*\right\|_\infty > \sigma\sqrt{\frac{k\log(k^2n)}{n}}\right) \le 2k\exp\left(-\frac{\sigma^2 k\log(k^2n)/n}{2k\sigma^2/n}\right) = \frac{2}{\sqrt{n}}.$$
Hence, with probability at least $1 - \frac{2}{\sqrt{n}}$, we have $\left\|\frac{k}{n}\widetilde{\boldsymbol{X}}^T\boldsymbol{y} - \tilde{\boldsymbol{\beta}}^*\right\|_\infty \le \sigma\sqrt{\frac{k\log(k^2n)}{n}} = \lambda k$, due to our choice of $\lambda = \sigma\sqrt{\frac{\log(k^2n)}{kn}}$. Under $\left\|\frac{k}{n}\widetilde{\boldsymbol{X}}^T\boldsymbol{y} - \tilde{\boldsymbol{\beta}}^*\right\|_\infty \le \lambda k$, the following results hold.

- By $\tilde{\beta}_k^* = 0$ and

$$\left|\frac{k}{n}(\widetilde{\boldsymbol{X}}_k)^T\boldsymbol{y}\right| = \left|\frac{k}{n}(\widetilde{\boldsymbol{X}}_k)^T\boldsymbol{y} - \tilde{\beta}_k^*\right| \le \left\|\frac{k}{n}\widetilde{\boldsymbol{X}}^T\boldsymbol{y} - \tilde{\boldsymbol{\beta}}^*\right\|_\infty \le \lambda k,$$

we have $\check{\beta}_{\lambda,k}^{oracle} = S\left(\frac{k}{n}(\widetilde{\boldsymbol{X}}_k)^T\boldsymbol{y}, \lambda k\right) = 0$ and $\check{\beta}_{\lambda,\ell}^{oracle} = \check{\beta}_{\lambda,k}^{oracle} = \tilde{\beta}_k^* = \beta_\ell^*$ for all $\ell > \frac{k-1}{k}n$.

- For $j = 1, \ldots, k-1$, since $\left|\frac{k}{n}(\widetilde{\boldsymbol{X}}_j)^T\boldsymbol{y} - \tilde{\beta}_j^*\right| \le \lambda k$ and $\left|\tilde{\beta}_j^*\right| \ge \min_{i=1,\ldots,k-1}\left|\tilde{\beta}_i^*\right| > 2\lambda k$, we must have $\frac{k}{n}(\widetilde{\boldsymbol{X}}_j)^T\boldsymbol{y}$ and $\tilde{\beta}_j^*$ share the same sign. Moreover, we either have

$$\left|\frac{k}{n}(\widetilde{\boldsymbol{X}}_j)^T\boldsymbol{y}\right| \ge \left|\tilde{\beta}_j^*\right| > \lambda k$$

or $\left|\frac{k}{n}(\widetilde{\boldsymbol{X}}_j)^T\boldsymbol{y}\right| < \left|\tilde{\beta}_j^*\right|$ in which case $\left|\frac{k}{n}(\widetilde{\boldsymbol{X}}_j)^T\boldsymbol{y} - \tilde{\beta}_j^*\right| = \left|\left|\frac{k}{n}(\widetilde{\boldsymbol{X}}_j)^T\boldsymbol{y}\right| - \left|\tilde{\beta}_j^*\right|\right| = \left|\tilde{\beta}_j^*\right| - \left|\frac{k}{n}(\widetilde{\boldsymbol{X}}_j)^T\boldsymbol{y}\right| \le \lambda k$ and therefore

$$\left|\frac{k}{n}(\widetilde{\boldsymbol{X}}_j)^T\boldsymbol{y}\right| \ge \left|\tilde{\beta}_j^*\right| - \lambda k \ge 2\lambda k - \lambda k = \lambda k.$$

Thus, $\left|\frac{k}{n}(\widetilde{\boldsymbol{X}}_j)^T\boldsymbol{y}\right| > \lambda k$ for $j = 1, \ldots, k-1$. By definition of $\hat{\boldsymbol{\beta}}_\lambda^{oracle}$ and (B.2), for $\frac{j-1}{k}n < \ell \le \frac{j}{k}n$,

$$\hat{\beta}_{\lambda,\ell}^{oracle} = \check{\beta}_{\lambda,j}^{oracle} = S\left(\frac{k}{n}(\widetilde{\boldsymbol{X}}_j)^T\boldsymbol{y}, \lambda k\right) = \frac{k}{n}(\widetilde{\boldsymbol{X}}_j)^T\boldsymbol{y}\left(1 - \frac{\lambda k}{\left|\frac{k}{n}(\widetilde{\boldsymbol{X}}_j)^T\boldsymbol{y}\right|}\right)$$

which is of the same sign as $\tilde{\beta}_j^*$ (and the same sign as $\beta_\ell^*$).

In the above two bullet points, we have shown $\mathbb{S}_\pm(\hat{\boldsymbol{\beta}}_\lambda^{oracle}) = \mathbb{S}_\pm(\boldsymbol{\beta}^*)$ holds with probability at least $1 - \frac{2}{\sqrt{n}}$. Hence,

$$\liminf_{n\to\infty} \mathbb{P}\left(\mathbb{S}_\pm(\hat{\boldsymbol{\beta}}_\lambda^{oracle}) = \mathbb{S}_\pm(\boldsymbol{\beta}^*)\right) \ge \lim_{n\to\infty} 1 - \frac{2}{\sqrt{n}} = 1.$$

Since $\limsup_{n\to\infty} \mathbb{P}\left(\mathbb{S}_\pm(\hat{\boldsymbol{\beta}}_\lambda^{oracle}) = \mathbb{S}_\pm(\boldsymbol{\beta}^*)\right) = 1$, the limit for $\mathbb{P}\left(\mathbb{S}_\pm(\hat{\boldsymbol{\beta}}_\lambda^{oracle}) = \mathbb{S}_\pm(\boldsymbol{\beta}^*)\right)$ is 1. □

**Lemma 11.** *Suppose* $\boldsymbol{\varepsilon} \sim N_n(0, \sigma^2\boldsymbol{I}_n)$ *and* $\tilde{c} = \frac{1}{3}e^{(\pi/2+2)^{-1}}\sqrt{\frac{1}{4} + \frac{1}{\pi}}$. *Then*

$$\mathbb{P}\left(\max_{j=1,\ldots,n}|\epsilon_j| \le \frac{2\sigma}{\sqrt{3}}\sqrt{\log(2\tilde{c}n)}\right) \le \left(1 - \frac{1}{n}\right)^n.$$

*Proof.* Let $Z$ be a standard Gaussian variable. Theorem 2.1 of Côté et al. (2012) provides a lower bound for the Gaussian Q function (i.e., $\mathbb{P}(Z > z)$). Choosing $\kappa = \frac{3}{2}$ in their Theorem 2.1 yields

$$\mathbb{P}(Z > z) \geq \underbrace{\left(\frac{1}{3} e^{(\pi/2+2)^{-1}} \sqrt{\frac{1}{4} + \frac{1}{\pi}}\right)}_{\tilde{c}} e^{-\frac{3z^2}{4}}$$

where $\tilde{c} = \frac{1}{3} e^{(\pi/2+2)^{-1}} \sqrt{\frac{1}{4} + \frac{1}{\pi}}$ is independent of $z$. Since $\epsilon_1 = \sigma Z$, we have for any $\eta > 0$

$$\mathbb{P}(\epsilon_1 > \eta) \geq \tilde{c} e^{-\frac{3\eta^2}{4\sigma^2}} \quad \Rightarrow \quad \mathbb{P}(|\epsilon_1| > \eta) \geq 2\tilde{c} e^{-\frac{3\eta^2}{4\sigma^2}}.$$

Moreover,

$$\mathbb{P}\left(\max_{j=1,\ldots,n} |\epsilon_j| \leq \eta\right) = \left(\mathbb{P}\left(|\epsilon_1| \leq \eta\right)\right)^n = \left(1 - \mathbb{P}\left(|\epsilon_1| > \eta\right)\right)^n \leq \left(1 - 2\tilde{c} e^{-\frac{3\eta^2}{4\sigma^2}}\right)^n.$$

Plugging in $\eta = \frac{2\sigma}{\sqrt{3}} \sqrt{\log(2\tilde{c}n)}$ in the above inequality yields

$$\mathbb{P}\left(\max_{j=1,\ldots,n} |\epsilon_j| \leq \frac{2\sigma}{\sqrt{3}} \sqrt{\log(2\tilde{c}n)}\right) \leq \left(1 - \frac{1}{n}\right)^n.$$

□

**Proposition 4** (Failure of support recovery of lasso)**.** *Suppose* $\min_{i=1,\ldots,k-1} \left|\tilde{\beta}_i^*\right| \leq \sigma \sqrt{\frac{\log(2\tilde{c}(k-1)n/k)}{3}}$ *where* $\tilde{c} = \frac{1}{3} e^{(\pi/2+2)^{-1}} \sqrt{\frac{1}{4} + \frac{1}{\pi}}$. *The lasso fails to get high-probability signed support recovery:*

$$\limsup_{n\to\infty} \sup_{\lambda \geq 0} \mathbb{P}\left(\mathbb{S}_{\pm}(\hat{\boldsymbol{\beta}}_\lambda^{lasso}) = \mathbb{S}_{\pm}(\boldsymbol{\beta}^*)\right) \leq \frac{1}{e}.$$

*Proof.* The lasso solution can be simplified to $\hat{\boldsymbol{\beta}}_\lambda^{lasso} = S(\boldsymbol{y}, \lambda)$. Since $\beta_\ell^* \neq 0$ for $\ell \leq \frac{k-1}{k} n$ and $\beta_\ell^* = 0$ for $\ell > \frac{k-1}{k} n$, the following is a necessary condition for $\hat{\boldsymbol{\beta}}_\lambda^{lasso}$ to recover the correct signed support:

$$\exists\, \lambda \text{ s.t. } |y_\ell| > \lambda \text{ for } \ell \leq \frac{k-1}{k} n \text{ and } |y_\ell| \leq \lambda \text{ for } \ell > \frac{k-1}{k} n \quad \Leftrightarrow \quad \min_{\ell \leq \frac{k-1}{k} n} |y_\ell| > \max_{\ell > \frac{k-1}{k} n} |y_\ell|.$$

Define $\bar{i} := \underset{i=1,\dots,k-1}{\arg\min} \left|\tilde{\beta}^*_i\right|$ and $\mathcal{A} := \left\{ \max_{\ell > \frac{k-1}{k}n} |\epsilon_\ell| \le \frac{2\sigma}{\sqrt{3}} \sqrt{\log\left(\frac{2\tilde{c}(k-1)}{k}n\right)} \right\}$. Then

$$
\begin{aligned}
&\mathbb{P}\left(\mathbb{S}_\pm(\hat{\boldsymbol{\beta}}^{lasso}_\lambda) = \mathbb{S}_\pm(\boldsymbol{\beta}^*)\right) \\
\le &\mathbb{P}\left( \min_{\ell \le \frac{k-1}{k}n} |y_\ell| > \max_{\ell > \frac{k-1}{k}n} |y_\ell| \right) \\
\le &\mathbb{P}\left( \min_{\frac{\bar{i}-1}{k}n < \ell \le \frac{\bar{i}}{k}n} |y_\ell| > \max_{\ell > \frac{k-1}{k}n} |y_\ell| \right) \\
\le &\mathbb{P}\left( \left|\tilde{\beta}^*_{\bar{i}}\right| + \min_{\frac{\bar{i}-1}{k}n < \ell \le \frac{\bar{i}}{k}n} |\epsilon_\ell| > \max_{\ell > \frac{k-1}{k}n} |\epsilon_\ell| \right) \\
= &\mathbb{P}\left( \left|\tilde{\beta}^*_{\bar{i}}\right| + \min_{\frac{\bar{i}-1}{k}n < \ell \le \frac{\bar{i}}{k}n} |\epsilon_\ell| > \max_{\ell > \frac{k-1}{k}n} |\epsilon_\ell| \middle| \mathcal{A}^c \right) \cdot \mathbb{P}(\mathcal{A}^c) + \mathbb{P}\left( \left|\tilde{\beta}^*_{\bar{i}}\right| + \min_{\frac{\bar{i}-1}{k}n < \ell \le \frac{\bar{i}}{k}n} |\epsilon_\ell| > \max_{\ell > \frac{k-1}{k}n} |\epsilon_\ell| \middle| \mathcal{A} \right) \cdot \mathbb{P}(\mathcal{A}) \\
\le &\mathbb{P}\left( \left|\tilde{\beta}^*_{\bar{i}}\right| + \min_{\frac{\bar{i}-1}{k}n < \ell \le \frac{\bar{i}}{k}n} |\epsilon_\ell| > \frac{2\sigma}{\sqrt{3}} \sqrt{\log\left(\frac{2\tilde{c}(k-1)}{k}n\right)} \right) + \mathbb{P}(\mathcal{A}) \\
= &\left[ \mathbb{P}\left( |\epsilon_1| > \frac{2\sigma}{\sqrt{3}} \sqrt{\log\left(\frac{2\tilde{c}(k-1)}{k}n\right)} - \left|\tilde{\beta}^*_{\bar{i}}\right| \right) \right]^{\frac{n}{k}} + \mathbb{P}(\mathcal{A}) \quad \text{(by } \epsilon_i \text{ being } i.i.d.\text{)} \\
\le &\left[ \mathbb{P}\left( |\epsilon_1| > \frac{\sigma}{\sqrt{3}} \sqrt{\log\left(\frac{2\tilde{c}(k-1)}{k}n\right)} \right) \right]^{\frac{n}{k}} + \mathbb{P}(\mathcal{A}) \quad \left( \text{by } \left|\tilde{\beta}^*_{\bar{i}}\right| \le \sigma \sqrt{\frac{\log(2\tilde{c}(k-1)n/k)}{3}} \right) \\
\le &\left[ 2\exp\left( -\frac{1}{2\sigma^2} \cdot \frac{\sigma^2}{3} \log\left(\frac{2\tilde{c}(k-1)}{k}n\right) \right) \right]^{\frac{n}{k}} + \left(1 - \frac{k}{n}\right)^{\frac{n}{k}} \left(\text{by Chernoff ineq and Lemma 11}\right) \\
\le &2^{n/k} \exp\left( -\frac{n}{6k} \log\left(\frac{2\tilde{c}(k-1)}{k}n\right) \right) + \left(1 - \frac{k}{n}\right)^{\frac{n}{k}} \\
= &2^{n/k} \left(\frac{2\tilde{c}(k-1)}{k}n\right)^{-n/(6k)} + \left(1 - \frac{k}{n}\right)^{\frac{n}{k}} \\
= &\left(\frac{\tilde{c}(k-1)}{32k}n\right)^{-n/(6k)} + \left(1 - \frac{k}{n}\right)^{\frac{n}{k}}
\end{aligned}
$$

which holds for all $\lambda \ge 0$. In particular,

$$
\sup_{\lambda \ge 0} \mathbb{P}\left(\mathbb{S}_\pm(\hat{\boldsymbol{\beta}}^{lasso}_\lambda) = \mathbb{S}_\pm(\boldsymbol{\beta}^*)\right) \le \left(\frac{\tilde{c}(k-1)}{32k}n\right)^{-n/(6k)} + \left(1 - \frac{k}{n}\right)^{\frac{n}{k}}.
$$

Taking lim sup on both side yields

$$
\limsup_{n\to\infty} \sup_{\lambda \ge 0} \mathbb{P}\left(\mathbb{S}_\pm(\hat{\boldsymbol{\beta}}^{lasso}_\lambda) = \mathbb{S}_\pm(\boldsymbol{\beta}^*)\right) \le \lim_{n\to\infty} \left(\frac{\tilde{c}(k-1)}{32k}n\right)^{-n/(6k)} + \lim_{n\to\infty} \left(1 - \frac{k}{n}\right)^{\frac{n}{k}}
$$

$$= 0 + \frac{1}{e} = \frac{1}{e}.$$

□

## B.3 Consensus ADMM for Solving Problem (3.5)

### B.3.1 Derivation of Algorithm 5

The ADMM updates involve minimizing the augmented Lagrangian of the global consensus problem (3.6),

$$\begin{aligned}
&L_\rho(\boldsymbol{\beta}^{(1)}, \boldsymbol{\beta}^{(2)}, \boldsymbol{\beta}^{(3)}, \boldsymbol{\gamma}^{(1)}, \boldsymbol{\gamma}^{(2)}, \boldsymbol{\beta}, \boldsymbol{\gamma}; \boldsymbol{v}^{(1)}, \boldsymbol{v}^{(2)}, \boldsymbol{v}^{(3)}, \boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}) \\
&= \frac{1}{2n} \left\| \boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}^{(1)} \right\|_2^2 + \lambda\alpha \|\boldsymbol{\gamma}_{-r}^{(1)}\|_1 + \lambda(1-\alpha)\|\boldsymbol{\beta}^{(2)}\|_1 + 1_\infty\{\boldsymbol{\beta}^{(3)} = A\boldsymbol{\gamma}^{(2)}\} \\
&\quad + \sum_{i=1}^{3} \left( \boldsymbol{v}^{(i)T}(\boldsymbol{\beta}^{(i)} - \boldsymbol{\beta}) + \frac{\rho}{2} \|\boldsymbol{\beta}^{(i)} - \boldsymbol{\beta}\|_2^2 \right) + \sum_{j=1}^{2} \left( \boldsymbol{u}^{(j)T}(\boldsymbol{\gamma}^{(j)} - \boldsymbol{\gamma}) + \frac{\rho}{2} \|\boldsymbol{\gamma}^{(j)} - \boldsymbol{\gamma}\|_2^2 \right).
\end{aligned}$$

1. Update $\boldsymbol{\beta}^{(1)}$.

$$\boldsymbol{\beta}^{(1)k+1} := \underset{\boldsymbol{\beta}^{(1)} \in \mathbb{R}^p}{\arg\min} \left\{ \frac{1}{2n} \left\| \boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}^{(1)} \right\|_2^2 + \left\langle \boldsymbol{v}^{(1)k}, (\boldsymbol{\beta}^{(1)} - \boldsymbol{\beta}^k) \right\rangle + \frac{\rho}{2} \|\boldsymbol{\beta}^{(1)} - \boldsymbol{\beta}^k\|_2^2 \right\}.$$

Let $\boldsymbol{X} = \text{SVD}(\boldsymbol{U}, \boldsymbol{D}, \boldsymbol{V})$ be the singular value decomposition of $\boldsymbol{X}$, where $\boldsymbol{U} \in \mathbb{R}^{n \times n}$ contains left singular vectors in columns, $\boldsymbol{V} \in \mathbb{R}^{p \times p}$ contains right singular vectors in columns, and $\boldsymbol{D} \in \mathbb{R}^{n \times p}$ is a rectangular diagonal matrix with decreasing singular values on the diagonal. First order condition to the above problem gives us:

$$\begin{aligned}
&(\boldsymbol{X}^T\boldsymbol{X} + n\rho\boldsymbol{I}_p)\boldsymbol{\beta}^{(1)k+1} = \boldsymbol{X}^T\boldsymbol{y} + n\rho\boldsymbol{\beta}^k - n\boldsymbol{v}^{(1)k} \\
\Rightarrow\ &\boldsymbol{V}(\boldsymbol{D}^T\boldsymbol{D} + n\rho\boldsymbol{I}_p)\boldsymbol{V}^T\boldsymbol{\beta}^{(1)k+1} = \boldsymbol{X}^T\boldsymbol{y} + n\rho\boldsymbol{\beta}^k - n\boldsymbol{v}^{(1)k}
\end{aligned}$$

$$\Rightarrow \boldsymbol{\beta}^{(1)k+1} = \boldsymbol{V} diag\left(([\boldsymbol{D}^T\boldsymbol{D}]_{ii} + n\rho)^{-1}\right)\boldsymbol{V}^T\left(\boldsymbol{X}^T\boldsymbol{y} + n\rho\boldsymbol{\beta}^k - n\boldsymbol{\nu}^{(1)k}\right).$$

When $n \geq p$, we have

$$\boldsymbol{\beta}^{(1)k+1} = \widetilde{\boldsymbol{V}} diag\left(([\widetilde{\boldsymbol{D}}^T\widetilde{\boldsymbol{D}}]_{ii} + n\rho)^{-1}\right)\widetilde{\boldsymbol{V}}^T\left(\boldsymbol{X}^T\boldsymbol{y} + n\rho\boldsymbol{\beta}^k - n\boldsymbol{\nu}^{(1)k}\right). \tag{B.3}$$

When $n < p$, the SVD can be expressed in a compact form: $\boldsymbol{D} = (\widetilde{\boldsymbol{D}} : \boldsymbol{0})$ and $\boldsymbol{V} = (\widetilde{\boldsymbol{V}} : \widetilde{\boldsymbol{V}}_\perp)$ where $\widetilde{\boldsymbol{D}} \in \mathbb{R}^{n\times n}$ and $\widetilde{\boldsymbol{V}} \in \mathbb{R}^{p\times n}$ are from the compact SVD of $\boldsymbol{X}$, and $\widetilde{\boldsymbol{V}}_\perp \in \mathbb{R}^{p\times(p-n)}$. Thus,

$$\begin{aligned}
\boldsymbol{V} diag\left(([\boldsymbol{D}^T\boldsymbol{D}]_{ii} + n\rho)^{-1}\right)\boldsymbol{V}^T &= \left(\widetilde{\boldsymbol{V}} : \widetilde{\boldsymbol{V}}_\perp\right) diag\left(([\boldsymbol{D}^T\boldsymbol{D}]_{ii} + n\rho)^{-1}\right)\begin{pmatrix}\widetilde{\boldsymbol{V}}^T\\ \widetilde{\boldsymbol{V}}_\perp^T\end{pmatrix}\\
&= \widetilde{\boldsymbol{V}} diag\left(([\widetilde{\boldsymbol{D}}^T\widetilde{\boldsymbol{D}}]_{ii} + n\rho)^{-1}\right)\widetilde{\boldsymbol{V}}^T + \widetilde{\boldsymbol{V}}_\perp\widetilde{\boldsymbol{V}}_\perp^T/(n\rho)\\
&= \widetilde{\boldsymbol{V}} diag\left(([\widetilde{\boldsymbol{D}}^T\widetilde{\boldsymbol{D}}]_{ii} + n\rho)^{-1}\right)\widetilde{\boldsymbol{V}}^T + (\boldsymbol{I}_p - \widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{V}}^T)/(n\rho).
\end{aligned}$$

So when $n < p$,

$$\boldsymbol{\beta}^{(1)k+1} = \left[\widetilde{\boldsymbol{V}} diag\left(([\widetilde{\boldsymbol{D}}^T\widetilde{\boldsymbol{D}}]_{ii} + n\rho)^{-1}\right)\widetilde{\boldsymbol{V}}^T + (\boldsymbol{I}_p - \widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{V}}^T)/(n\rho)\right]\left(\boldsymbol{X}^T\boldsymbol{y} + n\rho\boldsymbol{\beta}^k - n\boldsymbol{\nu}^{(1)k}\right). \tag{B.4}$$

Since $\widetilde{\boldsymbol{V}} = \boldsymbol{V}$ when $n \geq p$ and $\boldsymbol{V}\boldsymbol{V}^T = \boldsymbol{I}_p$, we have (B.4) boil to (B.3) in that case.

2. Update $\boldsymbol{\beta}^{(2)}$.

$$\boldsymbol{\beta}^{(2)k+1} := \underset{\boldsymbol{\beta}^{(2)}\in\mathbb{R}^p}{\arg\min}\left\{\frac{\rho}{2}\left\|\boldsymbol{\beta}^{(2)} - \left(\boldsymbol{\beta}^k - \frac{1}{\rho}\boldsymbol{\nu}^{(2)k}\right)\right\|_2^2 + \lambda(1-\alpha)\|\boldsymbol{\beta}^{(2)}\|_1\right\}.$$

The solution is simply elementwise soft-thresholding:

$$\beta_\ell^{(2)k+1} = S\left(\beta_\ell^k - \frac{1}{\rho}v_\ell^{(2)}, \frac{\lambda(1-\alpha)}{\rho}\right) \quad \forall \ell = 1, \ldots, p.$$

3. Update $\boldsymbol{\gamma}^{(1)}$.

$$\boldsymbol{\gamma}^{(1)k+1} := \underset{\boldsymbol{\gamma}^{(1)}\in\mathbb{R}^{|\mathcal{T}|}}{\arg\min}\left\{\frac{\rho}{2}\left\|\boldsymbol{\gamma}^{(1)} - \left(\boldsymbol{\gamma}^k - \frac{1}{\rho}\boldsymbol{u}^{(1)k}\right)\right\|_2^2 + \lambda\alpha\|\boldsymbol{\gamma}_{-r}^{(1)}\|_1\right\}.$$

Since root $\gamma_r^{(1)}$ is not penalized, the solution is the following:

$$\gamma_\ell^{(1)k+1} = \begin{cases} S\left(\gamma_\ell^k - \frac{1}{\rho}u_\ell^{(1)k}, \frac{\lambda\alpha}{\rho}\right) & \text{if } \ell \in \{1, \ldots, |\mathcal{T}|\}\backslash\{r\} \\ \gamma_\ell^k - \frac{1}{\rho}u_\ell^{(1)k} & \text{if } \ell = r. \end{cases}$$

4. Joint update of $\boldsymbol{\beta}^{(3)}$ and $\boldsymbol{\gamma}^{(2)}$.

$$\begin{pmatrix} \boldsymbol{\beta}^{(3)k+1} \\ \boldsymbol{\gamma}^{(2)k+1} \end{pmatrix} := \underset{\boldsymbol{\beta}^{(3)} \in \mathbb{R}^p, \boldsymbol{\gamma}^{(2)} \in \mathbb{R}^{|\mathcal{T}|}}{\arg\min} \left\{ \left\| \boldsymbol{\beta}^{(3)} - \left(\boldsymbol{\beta}^k - \frac{1}{\rho}\boldsymbol{v}^{(3)k}\right) \right\|_2^2 + \left\| \boldsymbol{\gamma}^{(2)} - \left(\boldsymbol{\gamma}^k - \frac{1}{\rho}\boldsymbol{u}^{(2)k}\right) \right\|_2^2 \right\}$$

$$\text{s.t. } (\boldsymbol{I}_p : -\boldsymbol{A}) \begin{pmatrix} \boldsymbol{\beta}^{(3)} \\ \boldsymbol{\gamma}^{(2)} \end{pmatrix} = 0.$$

The solution is the projection of $\begin{pmatrix} \boldsymbol{\beta}^k \\ \boldsymbol{\gamma}^k \end{pmatrix} - \frac{1}{\rho}\begin{pmatrix} \boldsymbol{v}^{(3)k} \\ \boldsymbol{u}^{(2)k} \end{pmatrix}$ onto the null space of $(\boldsymbol{I}_p : -\boldsymbol{A})$. Let $(\boldsymbol{I}_p : -\boldsymbol{A}) = \text{SVD}(\cdot, , \boldsymbol{Q})$ where $\boldsymbol{Q} = (\widetilde{\boldsymbol{Q}} : \widetilde{\boldsymbol{Q}}_\perp) \in \mathbb{R}^{(p+|\mathcal{T}|):(p+|\mathcal{T}|)}$ contains all the right singular vectors in columns. So $\boldsymbol{I}_{p+|\mathcal{T}|} = \boldsymbol{Q}\boldsymbol{Q}^T = \widetilde{\boldsymbol{Q}}\widetilde{\boldsymbol{Q}}^T + \widetilde{\boldsymbol{Q}}_\perp\widetilde{\boldsymbol{Q}}_\perp^T$. Since $\widetilde{\boldsymbol{Q}}$ corresponds to non-zero singular values of $(\boldsymbol{I}_p : -\boldsymbol{A})$ by construction, we have $\widetilde{\boldsymbol{Q}}_\perp$ corresponds to the zero singular values, making itself an orthonormal basis for the null space of $(\boldsymbol{I}_p : -\boldsymbol{A})$. Thus,

$$\begin{aligned} \begin{pmatrix} \boldsymbol{\beta}^{(3)k+1} \\ \boldsymbol{\gamma}^{(2)k+1} \end{pmatrix} &= \widetilde{\boldsymbol{Q}}_\perp(\widetilde{\boldsymbol{Q}}_\perp^T\widetilde{\boldsymbol{Q}}_\perp)^{-1}\widetilde{\boldsymbol{Q}}_\perp^T \left[ \begin{pmatrix} \boldsymbol{\beta}^k \\ \boldsymbol{\gamma}^k \end{pmatrix} - \frac{1}{\rho}\begin{pmatrix} \boldsymbol{v}^{(3)k} \\ \boldsymbol{u}^{(2)k} \end{pmatrix} \right] \\ &= \widetilde{\boldsymbol{Q}}_\perp\widetilde{\boldsymbol{Q}}_\perp^T \left[ \begin{pmatrix} \boldsymbol{\beta}^k \\ \boldsymbol{\gamma}^k \end{pmatrix} - \frac{1}{\rho}\begin{pmatrix} \boldsymbol{v}^{(3)k} \\ \boldsymbol{u}^{(2)k} \end{pmatrix} \right] \\ &= \left(\boldsymbol{I}_{p+|\mathcal{T}|} - \widetilde{\boldsymbol{Q}}\widetilde{\boldsymbol{Q}}^T\right) \left[ \begin{pmatrix} \boldsymbol{\beta}^k \\ \boldsymbol{\gamma}^k \end{pmatrix} - \frac{1}{\rho}\begin{pmatrix} \boldsymbol{v}^{(3)k} \\ \boldsymbol{u}^{(2)k} \end{pmatrix} \right] \end{aligned}$$

5. Update global variables $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$.

$$\boldsymbol{\beta}^{k+1} := \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min} \sum_{i=1}^{3} \left\| \boldsymbol{\beta} - \left( \boldsymbol{\beta}^{(i)k+1} + \frac{1}{\rho}\boldsymbol{v}^{(i)k} \right) \right\|_2^2 = \bar{\boldsymbol{\beta}}^{k+1} + \frac{1}{\rho}\bar{\boldsymbol{v}}^k \tag{B.5}$$

$$\boldsymbol{\gamma}^{k+1} := \underset{\boldsymbol{\gamma}\in\mathbb{R}^{|\mathcal{T}|}}{\arg\min} \sum_{j=1}^{2} \left\| \boldsymbol{\gamma} - \left( \boldsymbol{\gamma}^{(j)k+1} + \frac{1}{\rho}\boldsymbol{u}^{(j)k} \right) \right\|_2^2 = \bar{\boldsymbol{\gamma}}^{k+1} + \frac{1}{\rho}\bar{\boldsymbol{u}}^k \tag{B.6}$$

where $\bar{\boldsymbol{\beta}}^k := \frac{\boldsymbol{\beta}^{(1)k}+\boldsymbol{\beta}^{(2)k}+\boldsymbol{\beta}^{(3)k}}{3}$, $\bar{\boldsymbol{v}}^k := \frac{\boldsymbol{v}^{(1)k}+\boldsymbol{v}^{(2)k}+\boldsymbol{v}^{(3)k}}{3}$, $\bar{\boldsymbol{\gamma}}^k := \frac{\boldsymbol{\gamma}^{(1)k}+\boldsymbol{\gamma}^{(2)k}}{2}$ and $\bar{\boldsymbol{u}}^k := \frac{\boldsymbol{u}^{(1)k}+\boldsymbol{u}^{(2)k}}{2}$.

6. Update dual variables.

$$\boldsymbol{v}^{(1)k+1} := \boldsymbol{v}^{(i)k} + \rho(\boldsymbol{\beta}^{(i)k+1} - \boldsymbol{\beta}^{k+1}) \quad \text{for } i = 1, 2, 3,$$

$$\boldsymbol{u}^{(1)k+1} := \boldsymbol{u}^{(j)k} + \rho(\boldsymbol{\gamma}^{(j)k+1} - \boldsymbol{\gamma}^{k+1}) \quad \text{for } j = 1, 2.$$

Similarly, averaging the updates for $u$ and the udpates for $v$ gives

$$\bar{\boldsymbol{v}}^{k+1} = \bar{\boldsymbol{v}}^k + \rho(\bar{\boldsymbol{\beta}}^{k+1} - \boldsymbol{\beta}^{k+1}) \tag{B.7}$$

$$\bar{\boldsymbol{u}}^{k+1} = \bar{\boldsymbol{u}}^k + \rho(\bar{\boldsymbol{\gamma}}^{k+1} - \boldsymbol{\gamma}^{k+1}) \tag{B.8}$$

Substituting (B.5) and (B.6) into (B.7) and (B.8) yields that $\bar{\boldsymbol{v}}^{k+1} = \bar{\boldsymbol{u}}^{k+1} = 0$ after the first iteration.

Using $\boldsymbol{\beta}^k = \bar{\boldsymbol{\beta}}^k$ and $\boldsymbol{\gamma}^k = \bar{\boldsymbol{\gamma}}^k$ in the above updates, the updates become Lines 9-16 of Algorithm 5. Next, we follow Section 3.3.1 in Boyd et al. (2011) to determine the termination criteria. We first write Problem (3.6) in the same form as Problem (3.1) in Boyd et al. (2011) which is presented below in typewriter font:

$$\min_{\mathtt{x},\mathtt{z}} \quad \{\mathtt{f(x)} + \mathtt{g(z)} \text{ s.t. } \mathtt{Ax} + \mathtt{Bz} = \mathtt{c}\}$$

where

$$\mathtt{A} = \boldsymbol{I}_{3p+2|\mathcal{T}|}, \mathtt{B} = -\begin{pmatrix} \boldsymbol{I}_p & 0 \\ \boldsymbol{I}_p & 0 \\ \boldsymbol{I}_p & 0 \\ 0 & \boldsymbol{I}_{|\mathcal{T}|} \\ 0 & \boldsymbol{I}_{|\mathcal{T}|} \end{pmatrix}, \mathtt{c} = 0, \mathtt{x} = \begin{pmatrix} \boldsymbol{\beta}^{(1)} \\ \boldsymbol{\beta}^{(2)} \\ \boldsymbol{\beta}^{(3)} \\ \boldsymbol{\gamma}^{(1)} \\ \boldsymbol{\gamma}^{(2)} \end{pmatrix} \text{ and } \mathtt{z} = \begin{pmatrix} \boldsymbol{\beta} \\ \boldsymbol{\gamma} \end{pmatrix}.$$

The primal and dual residuals are

$$\mathtt{r}^{k+1} = \mathtt{A}\mathtt{x}^{k+1} + \mathtt{B}\mathtt{z}^{k+1} - c = \begin{pmatrix} \boldsymbol{\beta}^{(1)k+1} - \boldsymbol{\beta}^{k+1} \\ \boldsymbol{\beta}^{(2)k+1} - \boldsymbol{\beta}^{k+1} \\ \boldsymbol{\beta}^{(3)k+1} - \boldsymbol{\beta}^{k+1} \\ \boldsymbol{\gamma}^{(1)k+1} - \boldsymbol{\gamma}^{k+1} \\ \boldsymbol{\gamma}^{(2)k+1} - \boldsymbol{\gamma}^{k+1} \end{pmatrix} \text{ and } \mathtt{s}^{k+1} = \rho \mathtt{A}^T \mathtt{B}(\mathtt{z}^{k+1} - \mathtt{z}^k) = \rho \begin{pmatrix} \boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^k \\ \boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^k \\ \boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^k \\ \boldsymbol{\gamma}^{k+1} - \boldsymbol{\gamma}^k \\ \boldsymbol{\gamma}^{k+1} - \boldsymbol{\gamma}^k \end{pmatrix}.$$

By Condition (3.12) in Boyd et al. (2011), the ADMM algorithm stops when both residuals are small. In our case, the termination criteria are the following.

1. The primal residual is small:

$$\sqrt{\sum_{i=1}^{3} ||\boldsymbol{\beta}^{(i)k} - \boldsymbol{\beta}^k||_2^2 + \sum_{j=1}^{2} \left\|\boldsymbol{\gamma}^{(j)k} - \boldsymbol{\gamma}^k\right\|_2^2}$$
$$\leq \sqrt{3p + 2|\mathcal{T}|} \cdot \epsilon^{abs} + \epsilon^{rel} \cdot \max\left\{ \sqrt{\sum_{i=1}^{3} ||\boldsymbol{\beta}^{(i)k}||_2^2 + \sum_{j=1}^{2} \left\|\boldsymbol{\gamma}^{(j)k}\right\|_2^2}, \sqrt{3\left\|\boldsymbol{\beta}^k\right\|_2^2 + 2\left\|\boldsymbol{\gamma}^k\right\|_2^2} \right\}.$$

2. The dual residual is small:

$$\rho \cdot \sqrt{3||\boldsymbol{\beta}^k - \boldsymbol{\beta}^{k-1}||_2^2 + 2||\boldsymbol{\gamma}^k - \boldsymbol{\gamma}^{k-1}||_2^2} \leq \sqrt{3p + 2|\mathcal{T}|} \cdot \epsilon^{abs} + \epsilon^{rel} \cdot \sqrt{\sum_{i=1}^{3} ||\boldsymbol{v}^{(i)k}||_2^2 + \sum_{j=1}^{2} ||\boldsymbol{u}^{(j)k}||_2^2}.$$

### B.3.2 Treatment of Intercept in Problem (3.5)

When an intercept $\beta_0$ is included in the least squares, Problem (3.5) becomes:

$$\min_{\substack{\beta_0\in\mathbb{R},\boldsymbol{\beta}\in\mathbb{R}^p,\boldsymbol{\gamma}\in\mathbb{R}^{|\mathcal{T}|}\\ \text{s.t.}\quad \boldsymbol{\beta}=\boldsymbol{A}\boldsymbol{\gamma}}} \left\{\frac{1}{2n}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}-\beta_0\|_2^2+\lambda\left(\alpha\|\boldsymbol{\gamma}_{-r}\|_1+(1-\alpha)\|\boldsymbol{\beta}\|_1\right)\right\}. \tag{B.9}$$

First-order coniditon of the solution $(\hat{\beta}_0,\hat{\boldsymbol{\beta}})$ yields that

$$\left.\frac{\partial\frac{1}{2n}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}-\beta_0\|_2^2}{\partial\beta_0}\right|_{(\beta_0,\boldsymbol{\beta})=(\hat{\beta}_0,\hat{\boldsymbol{\beta}})}=\frac{1}{n}\mathbf{1}_n^T(\mathbf{1}_n\hat{\beta}_0-(\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{\beta}}))=\frac{1}{n}(n\hat{\beta}_0-\mathbf{1}_n^T(\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{\beta}}))=0.$$

So $\hat{\beta}_0=\frac{1}{n}\mathbf{1}_n^T(\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{\beta}})$ where $\mathbf{1}_n\in\mathbb{R}^n$ is a column vector. Plugging $\hat{\beta}_0$ in Problem (B.9) yields

$$\min_{\substack{\boldsymbol{\beta}\in\mathbb{R}^p,\boldsymbol{\gamma}\in\mathbb{R}^{|\mathcal{T}|}\\ \text{s.t.}\quad \boldsymbol{\beta}=\boldsymbol{A}\boldsymbol{\gamma}}} \left\{\frac{1}{2n}\left\|\left(\boldsymbol{I}_n-\tfrac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right)\boldsymbol{y}-\left(\boldsymbol{I}_n-\tfrac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right)\boldsymbol{X}\boldsymbol{\beta}\right\|_2^2+\lambda\left(\alpha\|\boldsymbol{\gamma}_{-r}\|_1+(1-\alpha)\|\boldsymbol{\beta}\|_1\right)\right\}$$

which can now be solved using our consensus ADMM algorithm.

## B.4 Proof of Lemma 6

We first show existence of such $B^*$ by providing a feasible procedure to find $B^*$. Suppose $\boldsymbol{\beta}^*$ has at least two distinct values (otherwise $B^*=\{r\}$ trivially). Start with $B=\mathcal{L}(\mathcal{T})$ so that the first constraint is satisfied. If for siblings $u,v$ in $B$ such that the second constraint is violated, by construction $\beta_j^*=\beta_k^*$ for $j\in\mathcal{L}(\mathcal{T}_u)$ and $k\in\mathcal{L}(\mathcal{T}_v)$. So we replace $u,v$ in $B$ with their parent node. We repeat the above steps until the second constraint is satisfied, while holding the first constraint. Thus, $B$ satisfies the two requirements for $B^*$.

Suppose $B^*$ and $\tilde{B}^*$ are different aggregating sets for $\boldsymbol{\beta}^*$. Without loss of generality, suppose there exists $u\in\tilde{B}^*$ but $u\notin B^*$. Then $u$ is a descendant or an

ancestor of some nodes in $B^*$; for either case the second constraint will be violated. Thus, such $u$ does not exist and $\tilde{B}^* = B^*$.

The existence and uniqueness of $\mathcal{A}^*$ follow from the definition of support of $\boldsymbol{\beta}^*$.

## B.5 Proof of Theorem 6

We follow the proof strategy used in Theorem 1 of Lou et al. (2016) to prove this theorem. If $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}})$ is a solution to Problem (3.5), then we have

$$\frac{1}{2n}\left\|\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}\right\|_2^2 + \lambda\Omega(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}}) \leq \frac{1}{2n}\left\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\right\|_2^2 + \lambda\Omega(\boldsymbol{\beta}, \boldsymbol{\gamma})$$

for any $(\boldsymbol{\beta}, \boldsymbol{\gamma})$ such that $\boldsymbol{\beta} = \boldsymbol{A}\boldsymbol{\gamma}$, where $\Omega(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \alpha\left\|\boldsymbol{\gamma}_{-r}\right\|_1 + (1-\alpha)\left\|\boldsymbol{\beta}\right\|_1$. Let $(\boldsymbol{\beta}^*, \boldsymbol{\gamma}^*)$ be such that

$$\boldsymbol{\beta}^* = \boldsymbol{A}_{B^*}\tilde{\boldsymbol{\beta}}^* \quad \text{and} \quad \gamma_\ell^* = \begin{cases} \tilde{\beta}_\ell^* & \text{if } \ell \in B^* \\ 0 & \text{otherwise.} \end{cases}$$

Plugging in $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}^* + \boldsymbol{\varepsilon}$ and $(\boldsymbol{\beta}, \boldsymbol{\gamma}) = (\boldsymbol{\beta}^*, \boldsymbol{\gamma}^*)$, with some algebra we have

$$\frac{1}{2n}\left\|\boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{X}\boldsymbol{\beta}^*\right\|_2^2 + \lambda\Omega(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}}) \leq \lambda\Omega(\boldsymbol{\beta}^*, \boldsymbol{\gamma}^*) + \frac{1}{n}\boldsymbol{\varepsilon}^T\boldsymbol{X}\hat{\boldsymbol{\Delta}}^{(\beta^*)} \tag{B.10}$$

where $\hat{\boldsymbol{\Delta}}^{(\beta^*)} = \hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*$. By $\hat{\boldsymbol{\beta}} = \boldsymbol{A}\hat{\boldsymbol{\gamma}}$ and $\boldsymbol{\beta}^* = \boldsymbol{A}\boldsymbol{\gamma}^*$ (and writing $\hat{\boldsymbol{\Delta}}^{(\gamma^*)} = \hat{\boldsymbol{\gamma}} - \boldsymbol{\gamma}^*$),

$$\frac{1}{n}\boldsymbol{\varepsilon}^T\boldsymbol{X}\hat{\boldsymbol{\Delta}}^{(\beta^*)} = \frac{1}{n}\boldsymbol{\varepsilon}^T\boldsymbol{X}\boldsymbol{A}\hat{\boldsymbol{\Delta}}^{(\gamma^*)}.$$

Define $V_j := \frac{1}{\sqrt{n}}\boldsymbol{X}_j^T\boldsymbol{\varepsilon}$ for $j = 1, \ldots, p$ and $U_\ell := \frac{1}{\sqrt{n}}\boldsymbol{A}_\ell^T\boldsymbol{X}^T\boldsymbol{\varepsilon}$ for $\ell = 1, \ldots, |\mathcal{T}|$. Then

$$\frac{1}{n}\boldsymbol{\varepsilon}^T\boldsymbol{X}\hat{\boldsymbol{\Delta}}^{(\beta^*)} = \frac{1}{\sqrt{n}}\sum_{j=1}^{p} V_j\hat{\boldsymbol{\Delta}}_j^{(\beta^*)} \quad \text{and} \quad \frac{1}{n}\boldsymbol{\varepsilon}^T\boldsymbol{X}\boldsymbol{A}\hat{\boldsymbol{\Delta}}^{(\gamma^*)} = \frac{1}{\sqrt{n}}\sum_{\ell=1}^{|\mathcal{T}|} U_\ell\hat{\boldsymbol{\Delta}}_\ell^{(\gamma^*)}.$$

Moreover, for any $j = 1, \ldots, p$, there is a leaf $u_\ell \in \mathcal{T}$ such that $\boldsymbol{X}_j = \boldsymbol{X}\boldsymbol{A}_\ell$. Writing $\boldsymbol{V} = (V_1, \ldots, V_p)$ and $\boldsymbol{U} = (U_1, \ldots, U_{|\mathcal{T}|})$, we have $\|\boldsymbol{V}\|_\infty \leq \|\boldsymbol{U}\|_\infty$ hold with probability one.

We next bound $\frac{1}{n}\boldsymbol{\varepsilon}^T \boldsymbol{X}\hat{\boldsymbol{\Delta}}^{(\boldsymbol{\beta}^*)} = (1-\alpha)\frac{1}{n}\boldsymbol{\varepsilon}^T \boldsymbol{X}\hat{\boldsymbol{\Delta}}^{(\boldsymbol{\beta}^*)} + \alpha\frac{1}{n}\boldsymbol{\varepsilon}^T \boldsymbol{X}\boldsymbol{A}\hat{\boldsymbol{\Delta}}^{(\boldsymbol{\gamma}^*)}$ in absolute value, where $0 \le \alpha \le (1+p^{-1})^{-1}$.

$$\begin{aligned}
\left|\frac{1}{n}\boldsymbol{\varepsilon}^T \boldsymbol{X}\hat{\boldsymbol{\Delta}}^{(\boldsymbol{\beta}^*)}\right| &\le (1-\alpha)\left|\frac{1}{\sqrt{n}}\sum_{j=1}^{p} V_j \hat{\boldsymbol{\Delta}}_j^{(\boldsymbol{\beta}^*)}\right| + \alpha\left|\frac{1}{\sqrt{n}}\sum_{\ell=1}^{|\mathcal{T}|} U_\ell \hat{\boldsymbol{\Delta}}_\ell^{(\boldsymbol{\gamma}^*)}\right| \\
&\le (1-\alpha)\frac{1}{\sqrt{n}}\sum_{j=1}^{p} |V_j| \cdot |\hat{\boldsymbol{\Delta}}_j^{(\boldsymbol{\beta}^*)}| + \alpha\frac{1}{\sqrt{n}}\sum_{\ell=1}^{|\mathcal{T}|} |U_\ell| \cdot |\hat{\boldsymbol{\Delta}}_\ell^{(\boldsymbol{\gamma}^*)}| \\
&\le (1-\alpha)\frac{1}{\sqrt{n}}\|\boldsymbol{V}\|_\infty \|\hat{\boldsymbol{\Delta}}^{(\boldsymbol{\beta}^*)}\|_1 + \alpha\frac{1}{\sqrt{n}}\|\boldsymbol{U}\|_\infty \|\hat{\boldsymbol{\Delta}}^{(\boldsymbol{\gamma}^*)}\|_1 \\
&\le (1-\alpha)\frac{1}{\sqrt{n}}\|\boldsymbol{U}\|_\infty \|\hat{\boldsymbol{\Delta}}^{(\boldsymbol{\beta}^*)}\|_1 + \alpha\frac{1}{\sqrt{n}}\|\boldsymbol{U}\|_\infty \left(|\hat{\boldsymbol{\Delta}}_r^{(\boldsymbol{\gamma}^*)}| + \|\hat{\boldsymbol{\Delta}}_{-r}^{(\boldsymbol{\gamma}^*)}\|_1\right) \\
&\le (1-\alpha+p^{-1}\alpha)\frac{1}{\sqrt{n}}\|\boldsymbol{U}\|_\infty \|\hat{\boldsymbol{\Delta}}^{(\boldsymbol{\beta}^*)}\|_1 + 2\alpha\frac{1}{\sqrt{n}}\|\boldsymbol{U}\|_\infty \|\hat{\boldsymbol{\Delta}}_{-r}^{(\boldsymbol{\gamma}^*)}\|_1 \qquad \text{(B.11)}
\end{aligned}$$

where the last inequality follows from observing that $\hat{\boldsymbol{\Delta}}^{(\boldsymbol{\beta}^*)} = \boldsymbol{A}_{-r}\hat{\boldsymbol{\Delta}}_{-r}^{(\boldsymbol{\gamma}^*)} + \mathbf{1}_p \hat{\boldsymbol{\Delta}}_r^{(\boldsymbol{\gamma}^*)}$ and

$$\begin{aligned}
p|\hat{\boldsymbol{\Delta}}_r^{(\boldsymbol{\gamma}^*)}| = \|\mathbf{1}_p \hat{\boldsymbol{\Delta}}_r^{(\boldsymbol{\gamma}^*)}\|_1 &\le \|\hat{\boldsymbol{\Delta}}^{(\boldsymbol{\beta}^*)}\|_1 + \|\boldsymbol{A}_{-r}\hat{\boldsymbol{\Delta}}_{-r}^{(\boldsymbol{\gamma}^*)}\|_1 \text{ (by triangle inequality)} \\
&\le \|\hat{\boldsymbol{\Delta}}^{(\boldsymbol{\beta}^*)}\|_1 + |\!|\!|\boldsymbol{A}_{-r}|\!|\!|_1 \|\hat{\boldsymbol{\Delta}}_{-r}^{(\boldsymbol{\gamma}^*)}\|_1 \text{ (by definition of } |\!|\!|\cdot|\!|\!|_1) \\
&\le \|\hat{\boldsymbol{\Delta}}^{(\boldsymbol{\beta}^*)}\|_1 + p\|\hat{\boldsymbol{\Delta}}_{-r}^{(\boldsymbol{\gamma}^*)}\|_1.
\end{aligned}$$

When $\alpha \le (1+p^{-1})^{-1}$, we have $(1-\alpha) \ge p^{-1}\alpha$ and then $(1-\alpha+p^{-1}\alpha) \le 2(1-\alpha)$. Thus, (B.11) becomes

$$\left|\frac{1}{n}\boldsymbol{\varepsilon}^T \boldsymbol{X}\hat{\boldsymbol{\Delta}}^{(\boldsymbol{\beta}^*)}\right| \le 2(1-\alpha)\frac{1}{\sqrt{n}}\|\boldsymbol{U}\|_\infty \|\hat{\boldsymbol{\Delta}}^{(\boldsymbol{\beta}^*)}\|_1 + 2\alpha\frac{1}{\sqrt{n}}\|\boldsymbol{U}\|_\infty \|\hat{\boldsymbol{\Delta}}_{-r}^{(\boldsymbol{\gamma}^*)}\|_1 \qquad \text{(B.12)}$$

Since $\boldsymbol{\varepsilon} \sim N_n(0, \sigma^2 \boldsymbol{I}_n)$, $U_\ell \sim N\left(0, \frac{\|\boldsymbol{X}\boldsymbol{A}_\ell\|_2^2}{n}\sigma^2\right)$ for $\ell = 1, \ldots, |\mathcal{T}|$. By Lemma 6.2 of Bühlmann and van de Geer (2011), we have for $x > 0$

$$\mathbb{P}\left(\frac{\|\boldsymbol{U}\|_\infty}{2\sqrt{n}} > \frac{\|\boldsymbol{X}\boldsymbol{A}_\ell\|_2\,\sigma}{\sqrt{2}n}\sqrt{x + \log|\mathcal{T}|}\right) \le 2e^{-x}.$$

By the construction of $\mathcal{T}$, each internal node has at least 2 child nodes. To go up to the next level from the leaf nodes, only one node "survives" among its

siblings. For $\mathcal{T}$ with $p$ leaf nodes, there must be at most $p-1$ internal nodes where the maximum number is achieved when $\mathcal{T}$ is a full binary tree. Thus, $|\mathcal{T}| \leq 2p$. Moreover, $\max_{\ell=1,\ldots,|\mathcal{T}|} \|\boldsymbol{X}\boldsymbol{A}_\ell\|_2 \leq \left\|\boldsymbol{X}\mathbf{1}_p\right\|_2 = \sqrt{n}$ since $\boldsymbol{X}$ has non-negative elements. We therefore have

$$\mathbb{P}\left(\frac{\|\boldsymbol{U}\|_\infty}{2\sqrt{n}} > \nu\right) \leq 2e^{-x} \quad \text{for } \nu = \frac{\sigma}{\sqrt{2n}}\sqrt{x + \log 2p}.$$

Choosing $x = \log 2p$, we have $\nu = \sigma\sqrt{\frac{\log 2p}{n}}$ and $\|\boldsymbol{U}\|_\infty / \sqrt{n} \leq 2\nu$ hold with probability at least $1 - p^{-1}$. Thus, we have the following inequality holding with high probability:

$$\left|\frac{1}{n}\boldsymbol{\varepsilon}^T \boldsymbol{X}\hat{\boldsymbol{\Delta}}^{(\beta^*)}\right| \leq 4(1-\alpha)\nu\|\hat{\boldsymbol{\Delta}}^{(\beta^*)}\|_1 + 4\alpha\nu\|\hat{\boldsymbol{\Delta}}^{(\gamma^*)}_{-r}\|_1. \tag{B.13}$$

Let $\lambda \geq 8\nu$ and $0 \leq \alpha \leq (1+p^{-1})^{-1}$. By (B.10) and (B.13), the following holds with probability at least $1 - p^{-1}$:

$$\begin{aligned}
\frac{1}{2n}\left\|\boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{X}\boldsymbol{\beta}^*\right\|_2^2 &\leq \frac{1}{2}\lambda\Omega(\hat{\boldsymbol{\Delta}}^{(\beta^*)}, \hat{\boldsymbol{\Delta}}^{(\gamma^*)}) - \lambda\Omega(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}}) + \lambda\Omega(\boldsymbol{\beta}^*, \boldsymbol{\gamma}^*) \\
&\leq \frac{1}{2}\left(\lambda\Omega(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}}) + \lambda\Omega(\boldsymbol{\beta}^*, \boldsymbol{\gamma}^*)\right) - \lambda\Omega(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}}) + \lambda\Omega(\boldsymbol{\beta}^*, \boldsymbol{\gamma}^*) \quad \text{(by triangle inequality)} \\
&\leq \frac{3}{2}\lambda\Omega(\boldsymbol{\beta}^*, \boldsymbol{\gamma}^*) = \frac{3}{2}\lambda\left(\alpha\left\|\tilde{\boldsymbol{\beta}}^*\right\|_1 + (1-\alpha)\|\boldsymbol{\beta}^*\|_1\right).
\end{aligned}$$

## B.6 Proof of Corollary 1

The first statement follows immediately from observing that $\|\tilde{\boldsymbol{\beta}}^*\|_1 \leq M|B^*|$ and $\|\boldsymbol{\beta}^*\|_1 \leq M|\mathcal{A}^*|$.

To show the second statement, we start by showing that $\alpha = \frac{|\mathcal{A}^*|}{|\mathcal{A}^*|+|B^*|} \leq (1 + p^{-1})^{-1}$ for all $p$. Since $|B^*| \geq 1$ and $|\mathcal{A}^*| \leq p$, the following holds for all $p$:

$$|\mathcal{A}^*| \leq p \cdot |B^*| \quad \Leftrightarrow \quad |\mathcal{A}^*| + p^{-1}|\mathcal{A}^*| \leq |\mathcal{A}^*| + |B^*| \quad \Leftrightarrow \quad \frac{|\mathcal{A}^*|}{|\mathcal{A}^*| + |B^*|} \leq (1+p^{-1})^{-1}.$$

From Theorem 6 we have

$$\begin{aligned}
\frac{1}{n}\left\|\boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{X}\boldsymbol{\beta}^*\right\|_2^2 &\leq 3\lambda M\left(\alpha|B^*| + (1-\alpha)|\mathcal{A}^*|\right) \\
&= 24\sigma M\sqrt{\frac{\log 2p}{n}} \cdot \left(\alpha|B^*| + (1-\alpha)|\mathcal{A}^*|\right) \text{ (by plugging in } \lambda) \\
&= 48\sigma M\sqrt{\frac{\log 2p}{n}} \cdot \left(\frac{1}{|\mathcal{A}^*|} + \frac{1}{|B^*|}\right)^{-1} \text{ (by plugging in } \alpha) \\
&\leq 48\sigma M\sqrt{\frac{\log 2p}{n}} \cdot \min\left(|\mathcal{A}^*|, |B^*|\right).
\end{aligned}$$

This final inequality follows since, for any $a, b > 0$, $(1/a + 1/b)^{-1} \leq (1/a + 0)^{-1} = a$, which establishes by symmetry in $a$ and $b$ that $(1/a + 1/b)^{-1} \leq \min\{a, b\}$.

APPENDIX C

**APPENDIX FOR CHAPTER 4**

### C.0.1 LA-ADMM for Solving the Consensus Probelm in (4.5)

Let $(\cdot, \widetilde{\boldsymbol{D}}, \widetilde{\boldsymbol{V}}) \leftarrow \text{SVD}_{\text{compact}}(\log(\boldsymbol{X}))$ be the compact singular value decomposition of $\log(\boldsymbol{X})$, where $\widetilde{\boldsymbol{D}} \in \mathbb{R}^{\min(n,p)\times\min(n,p)}$ is a diagonal matrix with non-zero singular values on the diagonal, and $\widetilde{\boldsymbol{V}} \in \mathbb{R}^{p\times\min(n,p)}$ contains the right singular vectors corresponding to these singular values. Similarly, in $(\cdot, \cdot, \widetilde{\boldsymbol{Q}}) \leftarrow \text{SVD}_{\text{compact}}\left(\begin{pmatrix} \boldsymbol{I}_p & -\boldsymbol{A} \\ \boldsymbol{1}^T & \boldsymbol{0}^T \end{pmatrix}\right)$, $\widetilde{\boldsymbol{Q}} \in \mathbb{R}^{(p+|\mathcal{T}|)\times(p+1)}$ columns correspond to the $p+1$ non-zero singular values.

Algorithm 9 has the LA-ADMM iterations, in which the conventional ADMM updates are called $N_{inner}$ times. The conventional ADMM is summarized in Algorithm 5. We derive the updates in Algorithm 5 below.

The conventional ADMM involves minimizing the augmented Lagrangian of (4.5),

$$
\begin{aligned}
&L_\rho(\boldsymbol{\beta}^{(1)}, \boldsymbol{\beta}^{(2)}, \boldsymbol{\beta}^{(3)}, \boldsymbol{\gamma}^{(1)}, \boldsymbol{\gamma}^{(2)}, \boldsymbol{\beta}, \boldsymbol{\gamma}; \boldsymbol{v}^{(1)}, \boldsymbol{v}^{(2)}, \boldsymbol{v}^{(3)}, \boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}) \\
&= \frac{1}{2n}\left\|\boldsymbol{y} - \log(\boldsymbol{X})\boldsymbol{\beta}^{(1)}\right\|_2^2 + \lambda_1\|\boldsymbol{\gamma}^{(1)}_{-\text{root}}\|_1 + \lambda_2\|\boldsymbol{\beta}^{(2)}\|_1 + 1_\infty\left\{\begin{pmatrix} \boldsymbol{I}_p \\ \boldsymbol{1}^T \end{pmatrix}\boldsymbol{\beta}^{(3)} = \begin{pmatrix} \boldsymbol{A} \\ \boldsymbol{0}^T \end{pmatrix}\boldsymbol{\gamma}^{(2)}\right\} \\
&\quad + \sum_{i=1}^{3}\left(\boldsymbol{v}^{(i)T}(\boldsymbol{\beta}^{(i)} - \boldsymbol{\beta}) + \frac{\rho}{2}\|\boldsymbol{\beta}^{(i)} - \boldsymbol{\beta}\|_2^2\right) + \sum_{j=1}^{2}\left(\boldsymbol{u}^{(j)T}(\boldsymbol{\gamma}^{(j)} - \boldsymbol{\gamma}) + \frac{\rho}{2}\|\boldsymbol{\gamma}^{(j)} - \boldsymbol{\gamma}\|_2^2\right). \qquad \text{(C.1)}
\end{aligned}
$$

Let $(\cdot, \boldsymbol{D}, \boldsymbol{V}) \leftarrow \text{SVD}(\log(\boldsymbol{X}))$ be the full singular value decomposition of $\log(\boldsymbol{X})$, where $\boldsymbol{D} \in \mathbb{R}^{n\times p}$ is a rectangular matrix with singular values on the

diagonal and $\boldsymbol{V} \in \mathbb{R}^{p\times p}$ is the right singular matrix. Similarly, let $(\cdot,\cdot,\boldsymbol{Q}) \leftarrow \text{SVD}\left(\begin{pmatrix} \boldsymbol{I}_p & -\boldsymbol{A} \\ \boldsymbol{1}^T & \boldsymbol{0}^T \end{pmatrix}\right)$ where $\boldsymbol{Q} = \left(\widetilde{\boldsymbol{Q}} : \widetilde{\boldsymbol{Q}}_\perp\right) \in \mathbb{R}^{(p+|\mathcal{T}|)\times(p+|\mathcal{T}|)}$ is an orthogonal matrix. Following Yan and Bien (2018), we derive the ADMM updates below.

1. Update $\boldsymbol{\beta}^{(1)}$.

$$\boldsymbol{\beta}^{(1)k+1} := \underset{\boldsymbol{\beta}^{(1)}\in\mathbb{R}^p}{\arg\min}\left\{\frac{1}{2n}\left\|\boldsymbol{y} - \log(\boldsymbol{X})\boldsymbol{\beta}^{(1)}\right\|_2^2 + \boldsymbol{v}^{(1)k^T}(\boldsymbol{\beta}^{(1)} - \boldsymbol{\beta}^k) + \frac{\rho}{2}\|\boldsymbol{\beta}^{(1)} - \boldsymbol{\beta}^k\|_2^2\right\}.$$

By optimality condition of the problem

$$\begin{aligned}
&\left(\log(\boldsymbol{X})^T\log(\boldsymbol{X}) + n\rho\boldsymbol{I}_p\right)\boldsymbol{\beta}^{(1)k+1} = \log(\boldsymbol{X})^T\boldsymbol{y} + n\rho\boldsymbol{\beta}^k - n\boldsymbol{v}^{(1)k} \\
\Rightarrow\ &\boldsymbol{V}\left(\boldsymbol{D}^T\boldsymbol{D} + n\rho\boldsymbol{I}_p\right)\boldsymbol{V}^T\boldsymbol{\beta}^{(1)k+1} = \log(\boldsymbol{X})^T\boldsymbol{y} + n\rho\boldsymbol{\beta}^k - n\boldsymbol{v}^{(1)k} \\
\Rightarrow\ &\boldsymbol{\beta}^{(1)k+1} = \boldsymbol{V}\mathbf{diag}\left(([\boldsymbol{D}^T\boldsymbol{D}]_{ii} + n\rho)^{-1}\right)\boldsymbol{V}^T\left(\log(\boldsymbol{X})^T\boldsymbol{y} + n\rho\boldsymbol{\beta}^k - n\boldsymbol{v}^{(1)k}\right)
\end{aligned}$$

When $n \geq p$, we have $\boldsymbol{V} = \widetilde{\boldsymbol{V}}$ and $\boldsymbol{D}^T\boldsymbol{D} = \widetilde{\boldsymbol{D}}^T\widetilde{\boldsymbol{D}}$. Thus,

$$\boldsymbol{\beta}^{(1)k+1} = \widetilde{\boldsymbol{V}}\mathbf{diag}\left(\left([\widetilde{\boldsymbol{D}}^T\widetilde{\boldsymbol{D}}]_{ii} + n\rho\right)^{-1}\right)\widetilde{\boldsymbol{V}}^T\left(\log(\boldsymbol{X})^T\boldsymbol{y} + n\rho\boldsymbol{\beta}^k - n\boldsymbol{v}^{(1)k}\right).$$

When $n < p$, we have $\boldsymbol{D} = \left(\widetilde{\boldsymbol{D}} : \boldsymbol{0}\right)$ and $\boldsymbol{V} = \left(\widetilde{\boldsymbol{V}} : \widetilde{\boldsymbol{V}}_\perp\right)$. Thus,

$$\begin{aligned}
\boldsymbol{V}\mathbf{diag}\left(([\boldsymbol{D}^T\boldsymbol{D}]_{ii} + n\rho)^{-1}\right)\boldsymbol{V}^T &= \left(\widetilde{\boldsymbol{V}} : \widetilde{\boldsymbol{V}}_\perp\right)\mathbf{diag}\left(([\boldsymbol{D}^T\boldsymbol{D}]_{ii} + n\rho)^{-1}\right)\begin{pmatrix}\widetilde{\boldsymbol{V}}^T \\ \widetilde{\boldsymbol{V}}_\perp^T\end{pmatrix} \\
&= \widetilde{\boldsymbol{V}}\mathbf{diag}\left(\left([\widetilde{\boldsymbol{D}}^T\widetilde{\boldsymbol{D}}]_{ii} + n\rho\right)^{-1}\right)\widetilde{\boldsymbol{V}}^T + \widetilde{\boldsymbol{V}}_\perp\widetilde{\boldsymbol{V}}_\perp^T/(n\rho) \\
&= \widetilde{\boldsymbol{V}}\mathbf{diag}\left(\left([\widetilde{\boldsymbol{D}}^T\widetilde{\boldsymbol{D}}]_{ii} + n\rho\right)^{-1}\right)\widetilde{\boldsymbol{V}}^T + (\boldsymbol{I}_p - \widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{V}}^T)/(n\rho).
\end{aligned}$$

Finally,

$$\boldsymbol{\beta}^{(1)k+1} = \left[\widetilde{\boldsymbol{V}}\mathbf{diag}\left(\left([\widetilde{\boldsymbol{D}}^T\widetilde{\boldsymbol{D}}]_{ii} + n\rho\right)^{-1}\right)\widetilde{\boldsymbol{V}}^T + \left(\boldsymbol{I}_p - \widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{V}}^T\right)/(n\rho)\right]\left(\log(\boldsymbol{X})^T\boldsymbol{y} + n\rho\boldsymbol{\beta}^k - n\boldsymbol{v}^{(1)k}\right).$$

Note that when $n \geq p$, $\widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{V}}^T = \boldsymbol{V}\boldsymbol{V}^T = \boldsymbol{I}_p$ since $\boldsymbol{V}$ is an orthogonal matrix.

2. Update $\boldsymbol{\beta}^{(2)}$.

$$\boldsymbol{\beta}^{(2)k+1} := \underset{\boldsymbol{\beta}^{(2)} \in \mathbb{R}^p}{\arg\min} \left\{ \frac{\rho}{2} \left\| \boldsymbol{\beta}^{(2)} - (\boldsymbol{\beta}^k - \frac{1}{\rho} \boldsymbol{v}^{(2)k}) \right\|_2^2 + \lambda_2 \|\boldsymbol{\beta}^{(2)}\|_1 \right\}.$$

The solution is simply elementwise soft-thresholding:

$$\boldsymbol{\beta}^{(2)k+1} = \textbf{soft-threshold}_{\lambda_2/\rho} \left( \boldsymbol{\beta}^k - \boldsymbol{v}^{(2)k}/\rho \right).$$

3. Update $\boldsymbol{\gamma}^{(1)}$.

$$\boldsymbol{\gamma}^{(1)k+1} := \underset{\boldsymbol{\gamma}^{(1)} \in \mathbb{R}^{|\mathcal{T}|}}{\arg\min} \left\{ \frac{\rho}{2} \left\| \boldsymbol{\gamma}^{(1)} - (\boldsymbol{\gamma}^k - \frac{1}{\rho} \boldsymbol{u}^{(1)k}) \right\|_2^2 + \lambda_1 \|\boldsymbol{\gamma}^{(1)}_{-\text{root}}\|_1 \right\}.$$

Since root $\gamma^{(1)}_{\text{root}}$ is not penalized, the solution is elementwise soft-thresholding for the non-root coordinates:

$$\boldsymbol{\gamma}^{(1)k+1}_{-\text{root}} = \textbf{soft-threshold}_{\lambda_1/\rho} \left( \boldsymbol{\gamma}^k_{-\text{root}} - \boldsymbol{u}^{(1)k}_{-\text{root}}/\rho \right) \quad \text{and} \quad \gamma^{(1)k+1}_{\text{root}} = \gamma^k_{\text{root}} - u^{(1)k}_{\text{root}}/\rho.$$

4. Joint update of $\boldsymbol{\beta}^{(3)}$ and $\boldsymbol{\gamma}^{(2)}$.

$$\begin{pmatrix} \boldsymbol{\beta}^{(3)k+1} \\ \boldsymbol{\gamma}^{(2)k+1} \end{pmatrix} := \underset{\boldsymbol{\beta}^{(3)} \in \mathbb{R}^p, \boldsymbol{\gamma}^{(2)} \in \mathbb{R}^{|\mathcal{T}|}}{\arg\min} \left\{ \left\| \boldsymbol{\beta}^{(3)} - (\boldsymbol{\beta}^k - \frac{1}{\rho} \boldsymbol{v}^{(3)k}) \right\|_2^2 + \left\| \boldsymbol{\gamma}^{(2)} - (\boldsymbol{\gamma}^k - \frac{1}{\rho} \boldsymbol{u}^{(2)k}) \right\|_2^2 \right\}$$

$$\text{s.t.} \begin{pmatrix} \boldsymbol{I}_p & -\boldsymbol{A} \\ \boldsymbol{1}^T & \boldsymbol{0}^T \end{pmatrix} \begin{pmatrix} \boldsymbol{\beta}^{(3)} \\ \boldsymbol{\gamma}^{(2)} \end{pmatrix} = \boldsymbol{0}.$$

The solution is the projection of $\begin{pmatrix} \boldsymbol{\beta}^k \\ \boldsymbol{\gamma}^k \end{pmatrix} - \frac{1}{\rho} \begin{pmatrix} \boldsymbol{v}^{(3)k} \\ \boldsymbol{u}^{(2)k} \end{pmatrix}$ onto the null space of $\begin{pmatrix} \boldsymbol{I}_p & -\boldsymbol{A} \\ \boldsymbol{1}^T & \boldsymbol{0}^T \end{pmatrix}$. An orthonormal basis of the null space is made up of the $|\mathcal{T}| - 1$ columns of $\boldsymbol{Q}$ that correspond to zero singular values in the SVD of

$\begin{pmatrix} \boldsymbol{I}_p & -\boldsymbol{A} \\ \boldsymbol{1}^T & \boldsymbol{0}^T \end{pmatrix}$, which we denoted $\tilde{\boldsymbol{Q}}_\perp \in \mathbb{R}^{(p+|\mathcal{T}|):(|\mathcal{T}|-1)}$. Thus,

$$\begin{aligned}
\begin{pmatrix} \boldsymbol{\beta}^{(3)k+1} \\ \boldsymbol{\gamma}^{(2)k+1} \end{pmatrix} &= \widetilde{\boldsymbol{Q}}_\perp (\widetilde{\boldsymbol{Q}}_\perp^T \widetilde{\boldsymbol{Q}}_\perp)^{-1} \widetilde{\boldsymbol{Q}}_\perp^T \left[ \begin{pmatrix} \boldsymbol{\beta}^k \\ \boldsymbol{\gamma}^k \end{pmatrix} - \frac{1}{\rho} \begin{pmatrix} \boldsymbol{v}^{(3)k} \\ \boldsymbol{u}^{(2)k} \end{pmatrix} \right] \\
&= \widetilde{\boldsymbol{Q}}_\perp \widetilde{\boldsymbol{Q}}_\perp^T \left[ \begin{pmatrix} \boldsymbol{\beta}^k \\ \boldsymbol{\gamma}^k \end{pmatrix} - \frac{1}{\rho} \begin{pmatrix} \boldsymbol{v}^{(3)k} \\ \boldsymbol{u}^{(2)k} \end{pmatrix} \right] \\
&= \left( \boldsymbol{I}_{p+|\mathcal{T}|} - \widetilde{\boldsymbol{Q}} \widetilde{\boldsymbol{Q}}^T \right) \left[ \begin{pmatrix} \boldsymbol{\beta}^k \\ \boldsymbol{\gamma}^k \end{pmatrix} - \frac{1}{\rho} \begin{pmatrix} \boldsymbol{v}^{(3)k} \\ \boldsymbol{u}^{(2)k} \end{pmatrix} \right]
\end{aligned}$$

5. Update global variables $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$.

$$\boldsymbol{\beta}^{k+1} := \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min} \sum_{i=1}^{3} \left\| \boldsymbol{\beta} - (\boldsymbol{\beta}^{(i)k+1} + \frac{1}{\rho} \boldsymbol{v}^{(i)k}) \right\|_2^2 = \bar{\boldsymbol{\beta}}^{k+1} + \frac{1}{\rho} \bar{\boldsymbol{v}}^k \tag{C.2}$$

$$\boldsymbol{\gamma}^{k+1} := \underset{\boldsymbol{\gamma} \in \mathbb{R}^{|\mathcal{T}|}}{\arg\min} \sum_{j=1}^{2} \left\| \boldsymbol{\gamma} - (\boldsymbol{\gamma}^{(j)k+1} + \frac{1}{\rho} \boldsymbol{u}^{(j)k}) \right\|_2^2 = \bar{\boldsymbol{\gamma}}^{k+1} + \frac{1}{\rho} \bar{\boldsymbol{u}}^k \tag{C.3}$$

where $\bar{\boldsymbol{\beta}}^k := \frac{\boldsymbol{\beta}^{(1)k} + \boldsymbol{\beta}^{(2)k} + \boldsymbol{\beta}^{(3)k}}{3}$, $\bar{\boldsymbol{v}}^k := \frac{\boldsymbol{v}^{(1)k} + \boldsymbol{v}^{(2)k} + \boldsymbol{v}^{(3)k}}{3}$, $\bar{\boldsymbol{\gamma}}^k := \frac{\boldsymbol{\gamma}^{(1)k} + \boldsymbol{\gamma}^{(2)k}}{2}$ and $\bar{\boldsymbol{u}}^k := \frac{\boldsymbol{u}^{(1)k} + \boldsymbol{u}^{(2)k}}{2}$.

6. Update dual variables.

$$\begin{aligned}
\boldsymbol{v}^{(i)k+1} &:= \boldsymbol{v}^{(i)k} + \rho(\boldsymbol{\beta}^{(i)k+1} - \boldsymbol{\beta}^{k+1}) \quad \text{for } i = 1, 2, 3, \\
\boldsymbol{u}^{(j)k+1} &:= \boldsymbol{u}^{(j)k} + \rho(\boldsymbol{\gamma}^{(j)k+1} - \boldsymbol{\gamma}^{k+1}) \quad \text{for } j = 1, 2.
\end{aligned}$$

Similarly, averaging the updates for $\boldsymbol{u}$ and the udpates for $\boldsymbol{v}$ gives

$$\bar{\boldsymbol{v}}^{k+1} = \bar{\boldsymbol{v}}^k + \rho(\bar{\boldsymbol{\beta}}^{k+1} - \boldsymbol{\beta}^{k+1}) \tag{C.4}$$

$$\bar{\boldsymbol{u}}^{k+1} = \bar{\boldsymbol{u}}^k + \rho(\bar{\boldsymbol{\gamma}}^{k+1} - \boldsymbol{\gamma}^{k+1}) \tag{C.5}$$

Substituting (C.2) and (C.3) into (C.4) and (C.5) yields that $\bar{\boldsymbol{v}}^{k+1} = \bar{\boldsymbol{u}}^{k+1} = \boldsymbol{0}$ after the first iteration.

Using $\boldsymbol{\beta}^k = \bar{\boldsymbol{\beta}}^k$ and $\boldsymbol{\gamma}^k = \bar{\boldsymbol{\gamma}}^k$ in the above updates, the updates become Lines 6-14 of Algorithm 10.

**Algorithm 9** LA-ADMM $(\boldsymbol{y}, \log(\boldsymbol{X}), \boldsymbol{A}, n, p, |\mathcal{T}|, \lambda_1, \lambda_2, \boldsymbol{\beta}^0, \boldsymbol{\gamma}^0, \rho, N_{\text{inner}}, N_{\text{outer}})$

1: $(\cdot, \widetilde{\boldsymbol{D}}, \widetilde{\boldsymbol{V}}) \leftarrow \text{SVD}_{\text{compact}}(\log(\boldsymbol{X}))$
2: $(\cdot, \cdot, \widetilde{\boldsymbol{Q}}) \leftarrow \text{SVD}_{\text{compact}} \begin{pmatrix} \boldsymbol{I}_p & -\boldsymbol{A} \\ \boldsymbol{1}^T & \boldsymbol{0}^T \end{pmatrix}$
3: **for** $m = 1, \ldots, N_{\text{outer}}$ **do**
4: $\quad (\boldsymbol{\beta}^m, \boldsymbol{\gamma}^m) \leftarrow \text{ADMM}(\boldsymbol{y}, \log(\boldsymbol{X}), \widetilde{\boldsymbol{D}}, \widetilde{\boldsymbol{V}}, \widetilde{\boldsymbol{Q}}, n, p, |\mathcal{T}|, \lambda_1, \lambda_2, \boldsymbol{\beta}^{m-1}, \boldsymbol{\gamma}^{m-1}, \rho, N_{\text{inner}})$
5: $\quad \rho \leftarrow 2\rho$
6: **end for**
**Output:** $\boldsymbol{\beta}^{N_{\text{outer}}}, \boldsymbol{\gamma}^{N_{\text{outer}}}$

**Algorithm 10** ADMM $(\boldsymbol{y}, \log(\boldsymbol{X}), \widetilde{\boldsymbol{D}}, \widetilde{\boldsymbol{V}}, \widetilde{\boldsymbol{Q}}, n, p, |\mathcal{T}|, \lambda_1, \lambda_2, \boldsymbol{\beta}^0, \boldsymbol{\gamma}^0, \rho, N_{\text{inner}})$

1: $\boldsymbol{\beta}^{(i)0} \leftarrow \boldsymbol{\beta}^0 \quad \forall i = 1, 2, 3$
2: $\boldsymbol{\gamma}^{(j)0} \leftarrow \boldsymbol{\gamma}^0 \quad \forall j = 1, 2$
3: $\boldsymbol{v}^{(i)0} \leftarrow \boldsymbol{0} \in \mathbb{R}^p \quad \forall i = 1, 2, 3$
4: $\boldsymbol{u}^{(j)0} \leftarrow \boldsymbol{0} \in \mathbb{R}^{|\mathcal{T}|} \quad \forall j = 1, 2$
5: **for** $k = 1, \ldots, N_{\text{inner}}$ **do**
6: $\quad \boldsymbol{\beta}^{(1)k} \leftarrow \left[\widetilde{\boldsymbol{V}}\mathbf{diag}\left(\left([\widetilde{\boldsymbol{D}}^T\widetilde{\boldsymbol{D}}]_{ii} + n\rho\right)^{-1}\right)\widetilde{\boldsymbol{V}}^T + \frac{1}{n\rho}\left(\boldsymbol{I}_p - \widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{V}}^T\right)\right]\left(\log(\boldsymbol{X})^T\boldsymbol{y} + n\rho\boldsymbol{\beta}^{k-1} - n\boldsymbol{v}^{(1)k-1}\right)$
7: $\quad \boldsymbol{\beta}^{(2)k} \leftarrow \textbf{soft-threshold}_{\lambda_2/\rho}\left(\boldsymbol{\beta}^{k-1} - \boldsymbol{v}^{(2)k-1}/\rho\right)$
8: $\quad \boldsymbol{\gamma}^{(1)k}_{-\text{root}} \leftarrow \textbf{soft-threshold}_{\lambda_1/\rho}\left(\boldsymbol{\gamma}^{k-1}_{-\text{root}} - \boldsymbol{u}^{(1)k-1}_{-\text{root}}/\rho\right)$
9: $\quad \gamma^{(1)k}_{\text{root}} \leftarrow \gamma^{k-1}_{\text{root}} - u^{(1)k-1}_{\text{root}}/\rho$
10: $\quad \begin{pmatrix} \boldsymbol{\beta}^{(3)k} \\ \boldsymbol{\gamma}^{(2)k} \end{pmatrix} \leftarrow \left(\boldsymbol{I}_{p+|\mathcal{T}|} - \widetilde{\boldsymbol{Q}}\widetilde{\boldsymbol{Q}}^T\right)\left[\begin{pmatrix} \boldsymbol{\beta}^{k-1} \\ \boldsymbol{\gamma}^{k-1} \end{pmatrix} - \frac{1}{\rho}\begin{pmatrix} \boldsymbol{v}^{(3)k-1} \\ \boldsymbol{u}^{(2)k-1} \end{pmatrix}\right]$
11: $\quad \boldsymbol{\beta}^k \leftarrow (\boldsymbol{\beta}^{(1)k} + \boldsymbol{\beta}^{(2)k} + \boldsymbol{\beta}^{(3)k})/3$
12: $\quad \boldsymbol{\gamma}^k \leftarrow (\boldsymbol{\gamma}^{(1)k} + \boldsymbol{\gamma}^{(2)k})/2$
13: $\quad \boldsymbol{v}^{(i)k} \leftarrow \boldsymbol{v}^{(i)k-1} + \rho(\boldsymbol{\beta}^{(i)k} - \boldsymbol{\beta}^k) \quad \forall i = 1, 2, 3$
14: $\quad \boldsymbol{u}^{(j)k} \leftarrow \boldsymbol{u}^{(j)k-1} + \rho(\boldsymbol{\gamma}^{(j)k} - \boldsymbol{\gamma}^k) \quad \forall j = 1, 2.$
15: **end for**
**Output:** $\boldsymbol{\beta}^{N_{\text{inner}}}, \boldsymbol{\gamma}^{N_{\text{inner}}}$

### C.0.2 Treatment of Environemental Covariates in (4.4) and (4.2)

When including $\boldsymbol{W\eta}$ in the quadratic loss in (4.4), optimality coniditon for the solution $(\hat{\boldsymbol{\eta}}, \hat{\boldsymbol{\beta}})$ yields that

$$\begin{aligned}\frac{\partial \frac{1}{2n}\left\|\boldsymbol{y}-\boldsymbol{W\eta}-\log(\boldsymbol{X})\boldsymbol{\beta}\right\|_2^2}{\partial \boldsymbol{\eta}}\Bigg|_{(\boldsymbol{\eta},\boldsymbol{\beta})=(\hat{\boldsymbol{\eta}},\hat{\boldsymbol{\beta}})} &= \frac{1}{n}\boldsymbol{W}^T(\boldsymbol{W}\hat{\boldsymbol{\eta}}-(\boldsymbol{y}-\log(\boldsymbol{X})\hat{\boldsymbol{\beta}})) \\ &= \frac{1}{n}(\boldsymbol{W}^T\boldsymbol{W}\hat{\boldsymbol{\eta}}-\boldsymbol{W}^T(\boldsymbol{y}-\log(\boldsymbol{X})\hat{\boldsymbol{\beta}})) = \boldsymbol{0}.\end{aligned}$$

So, $\hat{\boldsymbol{\eta}} = (\boldsymbol{W}^T\boldsymbol{W})^{-1}\boldsymbol{W}^T(\boldsymbol{y}-\log(\boldsymbol{X})\hat{\boldsymbol{\beta}})$. Plugging $\hat{\boldsymbol{\eta}}$ in (4.4) yields

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p,\boldsymbol{\gamma}\in\mathbb{R}^{|\mathcal{T}|}}\left\{\frac{1}{2n}\left\|\boldsymbol{P}\boldsymbol{y}-\boldsymbol{P}\log(\boldsymbol{X})\boldsymbol{\beta}\right\|_2^2+\lambda_1\|\boldsymbol{\gamma}_{-\text{root}}\|_1+\lambda_2\|\boldsymbol{\beta}\|_1 \text{ s.t. } \boldsymbol{\beta}=\boldsymbol{A\gamma} \text{ and } \mathbf{1}^T\boldsymbol{\beta}=0\right\},$$

where $\boldsymbol{P} = \boldsymbol{I}_n - \boldsymbol{W}(\boldsymbol{W}^T\boldsymbol{W})^{-1}\boldsymbol{W}^T$. With the same procedure, when including $\boldsymbol{W\eta}$ in the quadratic loss in (4.2), we get $\hat{\boldsymbol{\eta}} = (\boldsymbol{W}^T\boldsymbol{W})^{-1}\boldsymbol{W}^T(\boldsymbol{y}-\log(\boldsymbol{Z})\hat{\boldsymbol{\beta}})$ and a centered problem,

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^{p^{\text{tax}}}}\left\{\frac{1}{2n}\|\boldsymbol{P}\boldsymbol{y}-\boldsymbol{P}\log(\boldsymbol{Z})\boldsymbol{\beta}\|_2^2+\lambda\|\boldsymbol{\beta}\|_1 \text{ s.t. } \mathbf{1}^T\boldsymbol{\beta}=0\right\}.$$

## BIBLIOGRAPHY


Aitchison, J. (1982). The statistical analysis of compositional data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 44(2):139–177.

Arnold, T. B. and Tibshirani, R. J. (2014). *genlasso: Path algorithm for generalized lasso problems*. R package version 1.3.

Bach, F. (2008). Exploring large feature spaces with hierarchical multiple kernel learning. In *Proceedings of the 21st International Conference on Neural Information Processing Systems*, NIPS'08, pages 105–112, USA. Curran Associates Inc.

Bach, F., Jenatton, R., Mairal, J., and Obozinski, G. (2012). Structured sparsity through convex optimization. *Statist. Sci.*, 27(4):450–468.

BaconShone, J. and Aitchison, J. (1984). Log contrast models for experiments with mixtures. *Biometrika*.

Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Img. Sci.*, 2(1):183–202.

Bernstein, D. S. (2009). *Matrix Mathematics: Theory, Facts, and Formulas (Second Edition)*. Princeton University Press.

Bertsekas, D. P. (1999). *Nonlinear Programming*. Athena Scientific, Belmont, MA.

Bien, J., Bunea, F., and Xiao, L. (2016). Convex banding of the covariance matrix. *J. Amer. Statist. Assoc.*, 111(514):834–845.

Bien, J., Taylor, J., and Tibshirani, R. (2013). A lasso for hierarchical interactions. *Ann. Statist.*, 41(3):1111–1141.

Bien, J. and Tibshirani, R. (2011). Hierarchical clustering with prototypes via minimax linkage. *Journal of the American Statistical Association*, 106 495:1075–1084.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, New York, NY.

Bühlmann, P. and van de Geer, S. (2011). *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer Publishing Company, Incorporated, 1st edition.

Cao, Y., Zhang, A., and Li, H. (2017). Microbial Composition Estimation from Sparse Count Data. *ArXiv e-prints*.

Caporaso, J. G., Kuczynski, J., Stombaugh, J., Bittinger, K., Bushman, F. D., Costello, E. K., Fierer, N., Pena, A. G., Goodrich, J. K., Gordon, J. I., Huttley, G. A., Kelley, S. T., Knights, D., Koenig, J. E., Ley, R. E., Lozupone, C. A., McDonald, D., Muegge, B. D., Pirrung, M., Reeder, J., Sevinsky, J. R., Turnbaugh, P. J., Walters, W. A., Widmann, J., Yatsunenko, T., Zaneveld, J., and Knight, R. (2010). QIIME allows analysis of high-throughput community sequencing data. *Nat. Methods*, 7(5):335–336.

Chaudhary, N., Sharma, A. K., Agarwal, P., Gupta, A., and Sharma, V. K. (2015). 16S classifier: a tool for fast and accurate taxonomic classification of 16S rRNA hypervariable regions in metagenomic datasets. *PLoS ONE*, 10(2):e0116106.

Chen, J., Bushman, F. D., Lewis, J. D., Wu, G. D., and Li, H. (2013). Structure-constrained sparse canonical correlation analysis with an application to microbiome data analysis. *Biostatistics*, 14(2):244–258.

Choi, N., Li, W., and Zhu, J. (2010). Variable selection with the strong heredity constraint and its oracle property. *J. Amer. Statist. Assoc.*, 105(489):354–364.

Chouldechova, A. and Hastie, T. (2015). Generalized additive model selection. *ArXiv e-prints*.

Côté, F. D., Psaromiligkos, I. N., and Gross, W. J. (2012). A Chernoff-type Lower Bound for the Gaussian Q-function. *ArXiv e-prints*.

Dalalyan, A. S., Hebiri, M., and Lederer, J. (2017). On the prediction performance of the lasso. *Bernoulli*, 23(1):552–581.

de la Cruz, R. and Kreft, J.-U. (2018). Geometric mean extension for data sets with zeros. *ArXiv e-prints*.

Devaraj, S., Hemarajata, P., and Versalovic, J. (2013). The human gut microbiome and body metabolism: implications for obesity and diabetes. *Clin. Chem.*, 59(4):617–628.

Feinerer, I. and Hornik, K. (2016). *wordnet: WordNet Interface*. R package version 0.1-11.

Feinerer, I. and Hornik, K. (2017). *tm: Text Mining Package*. R package version 0.7-1.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. Bradford Books.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.*, 3:1289–1305.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.

Greer, R. L., Dong, X., Moraes, A. C., Zielke, R. A., Fernandes, G. R., Peremyslova, E., Vasquez-Perez, S., Schoenborn, A. A., Gomes, E. P., Pereira, A. C., Ferreira, S. R., Yao, M., Fuss, I. J., Strober, W., Sikora, A. E., Taylor, G. A., Gulati, A. S., Morgun, A., and Shulzhenko, N. (2016). Akkermansia muciniphila mediates negative effects of IFN$\gamma$ on glucose metabolism. *Nat Commun*, 7:13329.

Guinot, F., Szafranski, M., Ambroise, C., and Samson, F. (2017). Learning the optimal scale for GWAS through hierarchical SNP aggregation. *ArXiv e-prints*.

Haris, A., Witten, D., and Simon, N. (2016). Convex modeling of interactions with strong heredity. *J. Comput. Graph. Statist.*, 25(4):981–1004.

Huang, A. (2008). Similarity measures for text document clustering. In *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*, pages 49–56.

Jacob, L., Obozinski, G., and Vert, J. (2009). Group lasso with overlap and graph lasso. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 433–440, USA. ACM.

Jenatton, R., Audibert, J.-Y., and Bach, F. (2011a). Structured variable selection with sparsity-inducing norms. *J. Mach. Learn. Res.*, 12:2777–2824.

Jenatton, R., Mairal, J., Obozinski, G., and Bach, F. (2010). Proximal methods for sparse hierarchical dictionary learning. In *Proceedings of the 27th Inter-*

*national Conference on International Conference on Machine Learning*, ICML'10, pages 487–494, USA. Omnipress.

Jenatton, R., Mairal, J., Obozinski, G., and Bach, F. (2011b). Proximal methods for hierarchical sparse coding. *J. Mach. Learn. Res.*, 12:2297–2334.

Karlsson, C. L., Onnerfalt, J., Xu, J., Molin, G., Ahrne, S., and Thorngren-Jerneck, K. (2012). The microbiota of the gut in preschool children with normal and excessive body weight. *Obesity (Silver Spring)*, 20(11):2257–2261.

Khabbazian, M., Kriebel, R., Rohe, K., and Ané, C. (2016). Fast and accurate detection of evolutionary shifts in ornsteinuhlenbeck models. *Methods in Ecology and Evolution*, 7(7):811–824.

Kim, S., Xing, E. P., et al. (2012). Tree-guided group lasso for multi-response regression with structured sparsity, with an application to eqtl mapping. *The Annals of Applied Statistics*, 6(3):1095–1117.

Levina, E., Rothman, A., and Zhu, J. (2008). Sparse estimation of large covariance matrices via a nested lasso penalty. *Ann. Appl. Stat.*, 2(1):245–263.

Ley, R. E., Peterson, D. A., and Gordon, J. I. (2006). Ecological and evolutionary forces shaping microbial diversity in the human intestine. *Cell*, 124(4):837–848.

Li, H. (2015). Microbiome, metagenomics, and high-dimensional compositional data analysis. 2:73–94.

Lim, M. and Hastie, T. (2015). Learning interactions via hierarchical group-lasso regularization. *J. Comput. Graph. Statist.*, 24(3):627–654.

Lin, W., Shi, P., Feng, R., and Li, H. (2014). Variable selection in regression with compositional covariates. *Biometrika*, 101:785–797.

Liu, X., Yu, S., Janssens, F., Glänzel, W., Moreau, Y., and De Moor, B. (2010). Weighted hybrid clustering by combining text mining and bibliometrics on a large-scale journal database. *J. Am. Soc. Inf. Sci. Technol.*, 61(6):1105–1119.

Lou, Y., Bien, J., Caruana, R., and Gehrke, J. (2016). Sparse partially linear additive models. *Journal of Computational and Graphical Statistics*, 25(4):1126–1140.

Matsen, F. A., Kodner, R. B., and Armbrust, E. V. (2010). pplacer: linear time maximum-likelihood and Bayesian phylogenetic placement of sequences onto a fixed reference tree. *BMC Bioinformatics*, 11:538.

McMurdie, P. J. and Holmes, S. (2013). phyloseq: An r package for reproducible interactive analysis and graphics of microbiome census data. *PLOS ONE*, 8(4):1–11.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *ArXiv e-prints*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Mohammad, S. M. and Turney, P. D. (2013). Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

Mukherjee, R., Pillai, N. S., and Lin, X. (2015). Hypothesis testing for high-dimensional sparse binary regression. *Ann. Statist.*, 43(1):352–381.

Nelder, J. A. (1977). A reformulation of linear models. *J. Roy. Statist. Soc. Ser. A*, 140(1):48–77.

Nesterov, Y. (2013). Gradient methods for minimizing composite functions. *Math. Program.*, 140(1):125–161.

Nguyen, N. P., Warnow, T., Pop, M., and White, B. (2016). A perspective on 16S rRNA operational taxonomic unit clustering using sequence similarity. *NPJ Biofilms Microbiomes*, 2:16004.

Nicholson, W. B., Bien, J., and Matteson, D. S. (2014). Hierarchical vector autoregression. *ArXiv e-prints*.

Noto, J. M. and Peek, R. M. (2017). The gastric microbiome, its interaction with Helicobacter pylori, and its potential role in the progression to stomach cancer. *PLoS Pathog.*, 13(10):e1006573.

Obozinski, G., Jacob, L., and Vert, J.-P. (2011). Group lasso with overlaps: the latent group lasso approach. Research report.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Qin, J., Li, Y., Cai, Z., Li, S., Zhu, J., Zhang, F., Liang, S., Zhang, W., Guan, Y., Shen, D., Peng, Y., Zhang, D., Jie, Z., Wu, W., Qin, Y., Xue, W., Li, J., Han, L., Lu, D., Wu, P., Dai, Y., Sun, X., Li, Z., Tang, A., Zhong, S., Li, X., Chen, W., Xu, R., Wang, M., Feng, Q., Gong, M., Yu, J., Zhang, Y., Zhang, M., Hansen, T., Sanchez, G., Raes, J., Falony, G., Okuda, S., Almeida, M., LeChatelier, E., Renault, P., Pons, N., Batto, J. M., Zhang, Z., Chen, H., Yang, R., Zheng, W., Li, S., Yang, H., Wang, J., Ehrlich, S. D., Nielsen, R., Pedersen, O., Kristiansen, K.,

and Wang, J. (2012). A metagenome-wide association study of gut microbiota in type 2 diabetes. *Nature*, 490(7418):55–60.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Radchenko, P. and James, G. M. (2010). Variable selection using adaptive nonlinear interaction structures in high dimensions. *J. Amer. Statist. Assoc.*, 105(492):1541–1553.

Randolph, T. W., Zhao, S., Copeland, W., Hullar, M., and Shojaie, A. (2015). Kernel-Penalized Regression for Analysis of Microbiome Data. *ArXiv e-prints*.

Ridenhour, B. J., Brooker, S. L., Williams, J. E., Van Leuven, J. T., Miller, A. W., Dearing, M. D., and Remien, C. H. (2017). Modeling time-series data from microbial communities. *ISME J*, 11(11):2526–2537.

Rothman, A., Levina, E., and Zhu, J. (2010). A new approach to Cholesky-based covariance regularization in high dimensions. *Biometrika*, 97(3):539.

Sankaran, K. and Holmes, S. P. (2017). Latent Variable Modeling for the Microbiome. *ArXiv e-prints*.

Santacruz, A., Collado, M. C., Garcia-Valdes, L., Segura, M. T., Martin-Lagos, J. A., Anjos, T., Marti-Romero, M., Lopez, R. M., Florido, J., Campoy, C., and Sanz, Y. (2010). Gut microbiota composition is associated with body weight, weight gain and biochemical parameters in pregnant women. *Br. J. Nutr.*, 104(1):83–92.

Schloss, P., L Westcott, S., Ryabin, T., R Hall, J., Hartmann, M., Hollister, E., Lesniewski, R., Oakley, B., Parks, D., Robinson, C., W Sahl, J., Stres, B.,

G Thallinger, G., Van Horn, D., and Weber, C. (2009). Introducing mothur: Open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Applied and environmental microbiology*, 75(23):7537–41.

Schmidt, M. and Murphy, K. (2010). Convex structure learning in log-linear models: beyond pairwise potentials. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, volume 9, pages 709–716, Italy. PMLR.

She, Y., Wang, Z., and Jiang, H. (0). Group regularized estimation under structural hierarchy. *J. Amer. Statist. Assoc.*, 0(ja):0–0.

Shi, P., Zhang, A., and Li, H. (2016). Regression analysis for microbiome compositional data. *Ann. Appl. Stat.*, 10(2):1019–1040.

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2013). A sparse-group lasso. *J. Comput. Graph. Statist.*, 22(2):231–245.

Tang, Y., Li, M., and Niclolae, D. L. (2016). Phylogenetic Dirichlet-multinomial model for microbiome data. *ArXiv e-prints*.

Tessler, M., Neumann, J. S., Afshinnekoo, E., Pineda, M., Hersch, R., Velho, L. F. M., Segovia, B. T., Lansac-Toha, F. A., Lemke, M., DeSalle, R., Mason, C. E., and Brugler, M. R. (2017). Large-scale differences in microbial biodiversity discovery between 16S amplicon and shotgun sequencing. *Sci Rep*, 7(1):6589.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B. Stat. Methodol.*, 58:267–288.

Tibshirani, R. J. and Taylor, J. (2011). The solution path of the generalized lasso. *Ann. Statist.*, 39(3):1335–1371.

Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *J. Optim. Theory Appl.*, 109(3):475–494.

Turlach, B. A., Venables, W. N., and Wright, S. (2005). Simultaneous variable selection. *Technometrics*, 47:349–363.

Turnbaugh, P. J., Hamady, M., Yatsunenko, T., Cantarel, B. L., Duncan, A., Ley, R. E., Sogin, M. L., Jones, W. J., Roe, B. A., Affourtit, J. P., Egholm, M., Henrissat, B., Heath, A. C., Knight, R., and Gordon, J. I. (2009). A core gut microbiome in obese and lean twins. *Nature*, 457(7228):480–484.

Villa, S., Rosasco, L., Mosci, S., and Verri, A. (2014). Proximal methods for the latent group lasso penalty. *Comput. Optim. Appl.*, 58(2):381–407.

Wallace, M. (2007). *Jawbone Java WordNet API*.

Wang, H., Lu, Y., and Zhai, C. (2010). Latent aspect rating analysis on review text data: A rating regression approach. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 783–792, New York, NY, USA. ACM.

Wang, Q., Garrity, G. M., Tiedje, J. M., and Cole, J. R. (2007). Naive Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy. *Appl. Environ. Microbiol.*, 73(16):5261–5267.

Wang, T. and Zhao, H. (2017a). A Dirichlet-tree multinomial regression model for associating dietary nutrients with gut microorganisms. *Biometrics*, 73(3):792–801.

Wang, T. and Zhao, H. (2017b). Structured subcomposition selection in regression and its application to microbiome data analysis. *Ann. Appl. Stat.*, 11(2):771–791.

Wu, G. D., Chen, J., Hoffmann, C., Bittinger, K., Chen, Y. Y., Keilbaugh, S. A., Bewtra, M., Knights, D., Walters, W. A., Knight, R., Sinha, R., Gilroy, E., Gupta, K., Baldassano, R., Nessel, L., Li, H., Bushman, F. D., and Lewis, J. D. (2011). Linking long-term dietary patterns with gut microbial enterotypes. *Science*, 334(6052):105–108.

Xia, F., Chen, J., Fung, W. K., and Li, H. (2013). A logistic normal multinomial regression model for microbiome compositional data analysis. *Biometrics*, 69(4):1053–1063.

Xu, Y., Liu, M., Lin, Q., and Yang, T. (2017). Admm without a fixed penalty parameter: Faster convergence with new adaptive penalization. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 1267–1277. Curran Associates, Inc.

Yan, X. and Bien, J. (2017). Hierarchical sparse modeling: A choice of two group lasso formulations. *Statist. Sci.*, 32(4):531–560.

Yan, X. and Bien, J. (2018). Rare Feature Selection in High Dimensions. *ArXiv e-prints*.

Yuan, M., Joseph, V., and Zou, H. (2009). Structured variable selection and estimation. *Ann. Appl. Stat.*, 3(4):1738–1757.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. Ser. B. Stat. Methodol.*, 68:49–67.

Zhai, J., Kim, J., Knox, K. S., Twigg, H. L., Zhou, H., and Zhou, J. J. (2018). Variance Component Selection With Applications to Microbiome Taxonomic Data. *Front Microbiol*, 9:509.

Zhang, T., Shao, M.-F., and Ye, L. (2012). 454 pyrosequencing reveals bacterial diversity of activated sludge from 14 sewage treatment plants. *The ISME Journal*, 6(6):1137–1147.

Zhang, X., Shen, D., Fang, Z., Jie, Z., Qiu, X., Zhang, C., Chen, Y., and Ji, L. (2013). Human gut microbiota changes reveal the progression of glucose intolerance. *PLoS ONE*, 8(8):e71108.

Zhao, P., Rocha, G., and Yu, B. (2009). The composite absolute penalties family for grouped and hierarchical variable selection. *Ann. Statist.*, 37(6A):3468–3497.